University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

Theses, Dissertations, & Student Research in Computer Electronics & Engineering

Electrical & Computer Engineering, Department of

Summer 7-30-2014

Performance Analysis of Mobile Ad-Hoc Routing Protocols by Varying Mobility, Speed and Network Load

Nilotpal Sarmah
University of Nebraska-Lincoln, nsarmah@unomaha.edu

PERFORMANCE ANALYSIS OF MOBILE AD-HOC ROUTING PROTOCOLS BY VARYING MOBILITY, SPEED AND NETWORK LOAD

by

Nilotpal Dev Sarmah

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Telecommunications Engineering

Under the Supervision of Professor Yaoqing Yang

Lincoln, Nebraska

August, 2014

PERFORMANCE ANALYSIS OF MOBILE AD-HOC ROUTING PROTOCOLS BY VARYING MOBILITY, SPEED AND NETWORK LOAD

Nilotpal Dev Sarmah, M.S.

University of Nebraska, 2014

Adviser: Yaoqing Yang

One of the most promising network that has emerged from the technology world is the mobile ad-hoc network or MANET. It is a type of multi-hop network. Wireless by nature, MANETs do not have a specific network infrastructure. It is a collection of wireless mobile devices that communicate with each other without the help of any third party backbone like a base-station or a router. It can be hard to imagine how every node in this type of network communicate with one another without having a router. In MANETs, nodes change locations with time, configure themselves and get the information transmitted from source to destination without the help of any router or base station. Hence, for efficient data transmission, it is critical to understand the type of routing that is being used by these networks. Since they have no specific routers to handle these tasks, it can be a monumental task for the nodes to efficiently determine a path to forward and route their packets when they are at constant motion. This research makes a comprehensive performance analysis of the various mobile ad-hoc routing protocols. Over 160 simulation scenarios have been conducted and as many as 6 performance parameters are analyzed and compared in three different scales of network to make it a comprehensive analysis.

Significant work is done in this area for more than a decade and researchers around the world have come up with a wide range of results. In this research, the results from previous work are taken into account for comparison and a wide analysis is made to carve out the most efficient routing algorithm under various mobility scenarios. All the major proactive and reactive routing protocols viz. Destination sequenced distance vector (DSDV), Optimized Link State Routing (OLSR), Dynamic Source Routing (DSR) and Ad-hoc On-demand Distance Vector (AODV) protocols are compared in three different phases - mobility, speed and network load. Simulation results show that dynamic source routing protocol (DSR) performs the best in small networks while ad-hoc on demand distance vector (AODV) routing protocol performs the best in medium and large networks. Although OLSR fails to cope with the level of AODV, it can be a superior protocol having demonstrated comparable performance to AODV and its proactive nature of routing packets.

ACKNOWLEDGEMENT

I would like to record my sincere thanks to each and every one who helped and encouraged me during my course of study at the UNL.

First of all, I thank God for giving me this wonderful opportunity to come abroad and pursue the American dream. I always wanted to come to the United States for the master's degree program and with my hard work; it's His blessings that makes me stand where I am today.

I would like to record my sincere thanks to my adviser, Dr. Yaoqing (Lamar) Yang, for his constant guidance and support throughout the completion of my master's thesis. I sincerely thank him for his time, patience, motivation and knowledge that enabled me to develop an understanding of the subject.

I would like to offer my sincere thanks to the other committee members, Dr. Hamid Sharif and Dr. Yi Qian for their support, encouragement and valuable comments. I have worked closely with Dr. Sharif in the early part of my master's program and it was a sheer pleasure.

I dedicate this master's thesis to my uncle, Dr. Ratul Dev Sarmah and aunt, Mrs. Janumoni Sarmah without whom, enrolling in a master's program in the United States two years back would have been impossible. They are my only family in the United States and they have given me everything that a student can ever imagine- my identity, a place to live, food to eat, constant motivation and unconditional support without which this master's thesis would not have been possible. I am forever indebted to the amount of love they have offered me like their own son. I am grateful to God that I have gotten a mentor like my uncle who has been guiding me in every phase of my life and supported me unconditionally in every other way. I extend my love and gratitude to them and offer sincere thanks for helping me complete my master's program. My cousins, Rohan and Eeshan have been constant source of motivation and support for me. With the guys around, I never felt that I needed friends in a new place. We shared a connection more than brothers and I am grateful to them for being so supportive in everything we did together. They played a major role in completion of my degree.

I am forever grateful to my mother, Arati Sarmah and my father, Dwipendra Dev Sarmah for raising me up with good values in life and supporting me all throughout my life. They have been a constant source of motivation even after I came abroad. Also, my brother, Utpal Dev Sarmah, who believed in me and always, kept motivating me throughout my master's program. Sincere thanks to them for their complete moral and spiritual support.

I would like to record my thanks all the Master's and Ph.D friends of my department, Sushanta Mohan Rakshit, Pradhumna Shrestha, Prabhat Dahal, Guman Singh Chauhan, Deepraj Vernekar, Naji Albakay and Priyankar Bhattacharjee for constantly helping me throughout my master's program. It was a pleasure sharing the journey with these wonderful people, sharing our moments of joy and frustrations together.

Last but not the least; I would like to thank all my friends in the United States and back home for constantly supporting me and motivating me in every phase of my life. I am blessed to have every one of you close to my heart and thank you for being there whenever I was in need of love and support.

Table of Contents

Table of Figures

List of Tables

Nomenclature

MANET	Mobile ad-hoc network
VANET	Vehicular ad-hoc network
TCP	Transmission control protocol
AODV	Ad-hoc on demand distance vector
DSDV	Destination-sequenced distance vector
DSR	Dynamic source routing
OLSR	Optimized Link State Routing
RREQ	Route Request
RREP	Route Reply
DSRP	Dynamic source routing protocol
MPR	Multi-point relay
PDF	Packet delivery fraction
E2E	End-to-end
CBR	Constant Bit Rate
NRL	Normalized routing load
MAC	Medium access control

NS	Network simulator
OTCL	Object oriented tool command language

Chapter 1. Introduction to MANETs

1.1 Background and Motivation

A multi-hop network is a type of wireless network that uses more than one wireless node to transmit its information from a source node to a destination node. These nodes freely and dynamically self-organize themselves allowing them to interconnect seamlessly within a specific range. This concept is around for close to 20 years now and currently applied in various consumer electronics and military applications. The concept evolved from single-hop networks where the information is transmitted through a single hop. One of the most common single-hop networks is the Bluetooth Piconet where two nodes can seamlessly transmit information to one another if they are within the transmission range.

Mobile ad-hoc network (MANET) is a type of multi-hop network. In this type of network, each node is free to move independently in any direction and hence the nodes change their links frequently. MANET has been a popular research topic since mid-1990. In contrast to conventional cellular networks, there is no master-slave relationship between the base station and the mobile users. MANETs is used in several applications like vehicular communications, military applications, emergency first response and public safety response.

Another type of mobile ad-hoc network is called the multi-hop cellular networks. As the name suggests, it is a type of cellular network that deploys multi-hop unlike the single-hop between the base station and the mobile users in conventional cellular networks.

Multi-hop cellular networks avoid the problem of fixed bases in single-hop cellular networks. Results have shown that multi-hop demonstrate significant improve in the throughput and overall efficiency when compared to single-hop networks. A key feature of the multi-hop cellular networks is that mobile stations can directly communicate with each other if they are mutually reachable unlike in single hop cellular network. This type of network is highly suited for use in situations where a fixed infrastructure is not available [1]. These types of networks are widely applied to consumer and military applications.

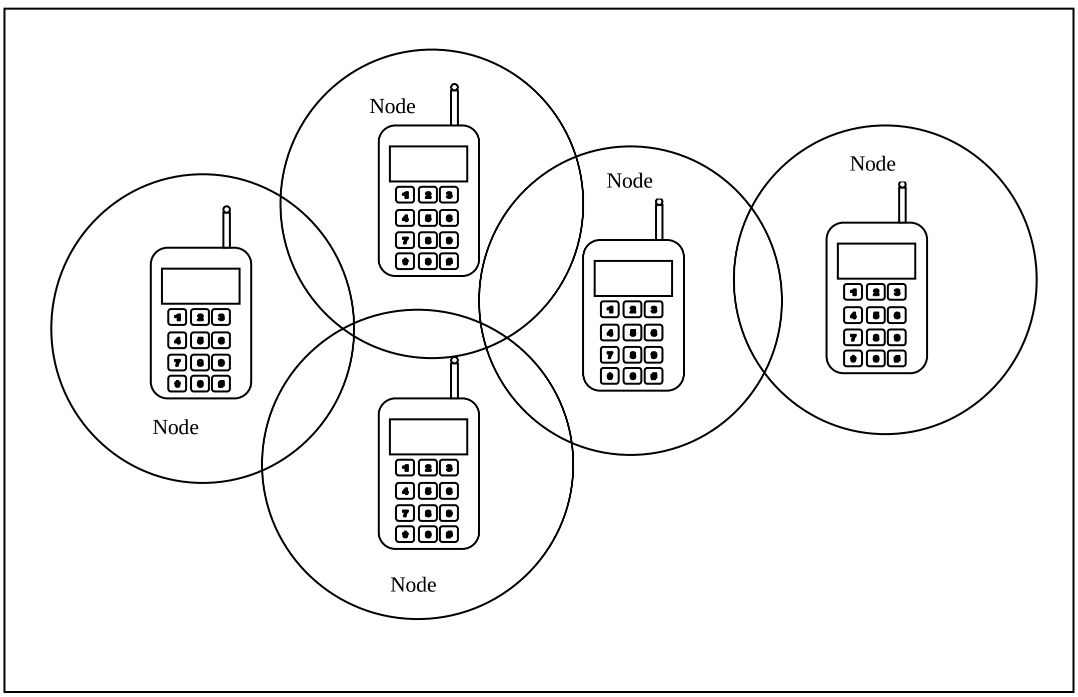

Figure 1.1 The Mobile ad-hoc concept

In figure 1.1, a mobile ad-hoc concept is presented. Node 1 to Node 5 is different mobile nodes which can communicate to each other independently. The circles around the nodes depict the wireless transmission range of the nodes. As it is seen from the figure, Node 3 cannot communicate directly with Node 5 since the transmission range of these nodes does not overlap each other. However, it can communicate with Node 5 via Node 1, 2 or 4 since their respective wireless ranges overlap with range of 3 and 5.

1.2 Benefits and applications of MANET

MANETs have several benefits. Unlike single hop networks which are bound to a certain range between the source and the receiving nodes, they can be extended to a wide range, thus extending the overall coverage of the network. Since the transmission is carried out over short links, the transmission power and energy is usually less. They enable higher rates resulting in higher throughput and more efficient use of the wireless medium. MANETs also avoid wide deployment of cables and the transmission can be carried out in a cost effective way.

VANETs or Vehicular ad-hoc network communications are one of the new challenging application areas for MANETs, and vehicle collision warning is one of the very promising potential applications in this field, since traffic accidents cause hundreds of thousands of fatalities and injuries every year. The results and simulations of MANET can be applied to VANET considering the fact that VANET is an application of MANET.

In Figure 1.2, a VANET is presented where vehicles can communicate with each other with different Remote Subscriber Units (RSU) and a Wimax base station. Three kinds of communications are shown in this figure - vehicle-to-vehicle, vehicle-to-roadside and inter-road communications. With various transmission ranges available, each of the node can communicate with each other even though their moving. The vehicles self-configure themselves based on the location of their nearest node and can transmit information from a certain source to destination.

Due to the life-critical nature of emergency applications, however, it is essential to ensure the solutions to be deployed work with almost 100% success rate, thus meeting the high standards required, even under extremely unfavorable conditions.

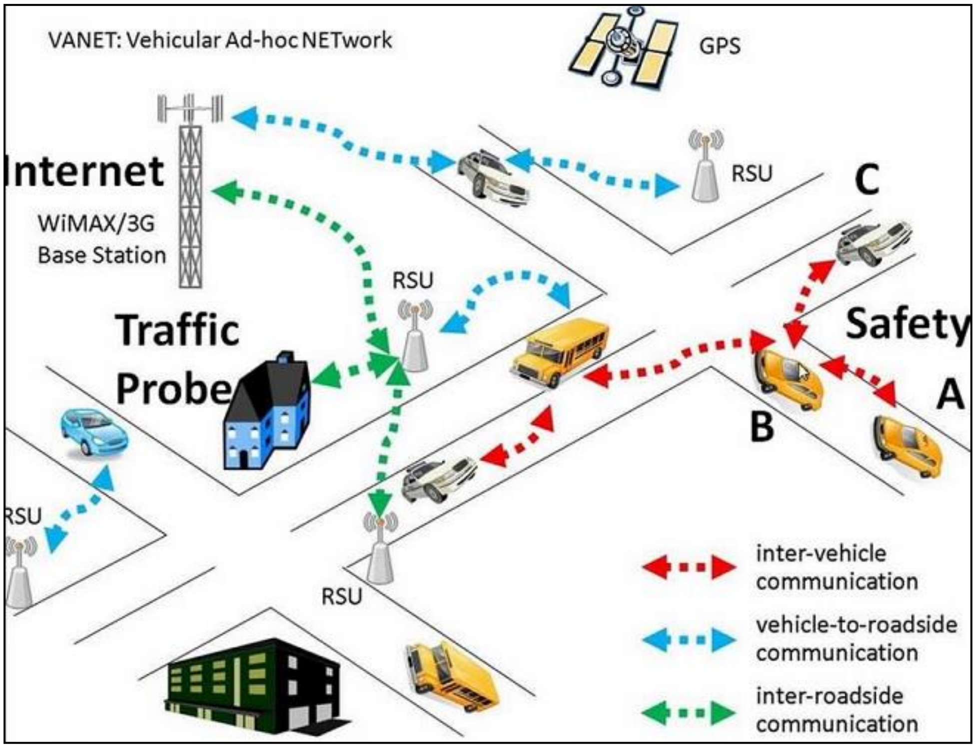

Figure 1.2 The VANET [2]

In sparse networks, for instance, where node connectivity is low, message dissemination becomes very difficult, and it is necessary to take additional measures in order to keep all nodes informed.

1.3 TCP and its performance in MANET

TCP (Transmission Control Protocol) is a set of rules (protocol) used along with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet. TCP is known as a connection-oriented protocol. TCP is responsible for ensuring that a message is divided into the packets that IP manages and for reassembling the packets back into the complete message at the other end. Transmission Control Protocol (TCP) was designed for reliable communication in computer networks. At the time it was conceptualized the computer networks were wired and hence the guiding principles of the design were in keeping with the characteristics of a wired network. Since then wireless networks have gained in popularity and is now all pervasive. Wireless networks are inherently more error-prone than wired networks due to several channel characteristics. The effects of fading, multipath etc lead to higher errors and packet losses in a wireless environment. TCP was designed to infer packet losses as a sign of network congestion and take corrective measures accordingly. In wireless networks this inference is wrongly made even when the loss of a packet or error in transmission is due to channel losses and not congestion. This leads to excessive number of retransmissions and timeout events leading to exponential decay of network performance in a very short period of time. Research has been conducted to lead to modifications to the TCP design to cater to the specific requirements of a wireless environment. There is a plethora of ways that have been suggested to mitigate the effects of wireless channel on TCP. When TCP was designed, certain routing algorithms were designed to control the traffic flow and optimize the network performance. However, with evolving wireless networks, those routing algorithms have failed to provide optimum network performance because they were designed to deal with congestion and ways to prevent congestion. Hence these routing algorithms were proactive in nature. It means they would take an action after the problem had occurred. However, with wireless scenarios, routing protocols are needed that were reactive in nature. It means changing the ongoing routing table instantly whenever there is congestion or packet loss due to link contention or any external scenario. In multi-hop networks, since there are no intermediate routers to route packets, the nodes have to efficiently determine the path to send the information from source to destination. Thus performance of the network highly depends on the efficiency of the routing protocol which in turn affects performance of TCP.

In recent years, researchers have proposed many schemes to improve performance of TCP in multi-hop wireless networks. TCP congestion control mechanisms are based on the fact that the main reason for loss is the buffer overflow. This mechanism is not adapted with ad-hoc networks where the main reason for loss is link contention caused by hidden terminal problems.

In this thesis, a comprehensive performance analysis of the mobile ad-hoc routing protocols is carried out and towards the end; a conclusion is drawn as to which routing protocol could be deemed efficient in what kind of an ad-hoc network scenario.

1.4 Proactive and Reactive Routing

Routing is defined as the process of finding a path from source to a destination. Mobile ad hoc networks, or MANET, are fundamentally different from traditional wired networks as wired networks are assumed to be stationary and static. So the routing protocols designed for wired networks can't work efficiently in mobile ad-hoc networks. This imposes different design requirement and constraints on routing protocols for

MANET. [3] A number of routing protocols have been suggested for ad-hoc networks. These protocols can be classified into two main categories: Proactive (table-driven) and Reactive (source-initiated or demand-driven).

Proactive routing protocols *or table-driven protocols* follow an approach similar to the one used in wired routing protocols. By continuously evaluating the known routes and attempting to discover new routes, they try to maintain the most up-to-date map of the network. This allows them to efficiently forward packets, as the route is known at the time when the packet arrives at the node. Destination Sequenced Distance Vector (DSDV) and Optimized Link State Routing (OLSR) protocols are examples of proactive protocols.

In contrast to proactive routing, reactive routing *or on-demand protocols* does not attempt to continuously determine the network connectivity. Instead, a route determination procedure is invoked *on demand* when a packet needs to be forwarded. The technique relies on queries that are flooded throughout the network. Examples are Ad-hoc on-demand distance vector (AODV) and Dynamic Source Routing (DSR) protocols. In DSR and AODV, a reply is sent back to the query source along the reverse path that the query traveled. The main difference is that DSR performs source routing with the addresses obtained from the query packet, while AODV uses next hop information stored in the nodes of the route.

2.1.1 DSDV

The Destination-Sequenced Distance-Vector (DSDV) Routing Algorithm developed by C. Perkins and P. Bhagwat in 1994 is based on the idea of the classical Bellman-Ford

Routing Algorithm with certain improvements. [4]

Every mobile station maintains a routing table that lists all available destinations, the number of hops to reach the destination and the sequence number assigned by the destination node. The sequence number is used to distinguish stale routes from new ones and thus avoid the formation of loops. The stations periodically transmit their routing tables to their immediate neighbors. A station also transmits its routing table if a significant change has occurred in its table from the last update sent. So, the update is both time-driven and event-driven. The routing table updates can be sent in two ways:- a "full dump" or an incremental update. A full dump sends the full routing table to the neighbors and could span many packets whereas in an incremental update only those entries from the routing table are sent that has a metric change since the last update and it must fit in a packet. If there is space in the incremental update packet then those entries may be included whose sequence number has changed. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a fast-changing network, incremental packets can grow big so full dumps will be more frequent. Each route update packet, in addition to the routing table information, also contains a unique sequence number assigned by the transmitter. The route labeled with the highest (i.e. most recent) sequence number is used. If two routes have the same sequence number then the route with the best metric (i.e. shortest route) is used. Based on the past history, the stations estimate the settling time of routes. The stations delay the transmission of a routing update by settling time so as to eliminate those updates that would occur if a better route were found very soon. [4]

2.1.2 OLSR

Optimized Link State Routing (OLSR) protocol is a proactive routing protocol where the routes are always immediately available when needed. OLSR is an optimized version of a pure link state protocol in which the topological changes cause the flooding of the topological information to all available hosts in the network. OLSR may optimize the reactivity to topological changes by reducing the maximum time interval for periodic control message transmission. Furthermore, as OLSR continuously maintains routes to all destinations in the network, the protocol is beneficial for traffic patterns where a large subset of nodes are communicating with another large subset of nodes, and where the [source, destination] pairs are changing over time. OLSR protocol is well suited for the application which does not allow the long delays in the transmission of the data packets. The best working environment for OLSR protocol is a dense network, where the most communication is concentrated between a large numbers of nodes. OLSR reduce the control overhead forcing the multi-point relay (MPR) to propagate the updates of the link state, also the efficiency is gained compared to classical link state protocol when the selected MPR set is as small as possible. But the drawback of this is that it must maintain the routing table for all the possible routes, so there is no difference in small networks, but when the number of the mobile hosts increase, then the overhead from the control messages is also increasing. This constrains the scalability of the OLSR protocol. The OLSR protocol work most efficiently in the dense networks. [9]

2.1.3 AODV

The Ad Hoc On-Demand Distance Vector (AODV) routing protocol which improves from DSDV is a reactive routing protocol. AODV minimizes the number of required broadcasts by creating routes in an on-demand manner. When a source node desires to send data to other destination node, it needs to initiate a path discovery process to locate the other node. A source node broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until the destination is located [3].

Figure 1.3 shows the route discovery process of AODV. Node 1 which is the source is broadcasting its request (RREQ) to its nearest nodes 2,3 and 4 which in turn forward the request to the subsequent nodes 5,6,7 and the destination node, 8.

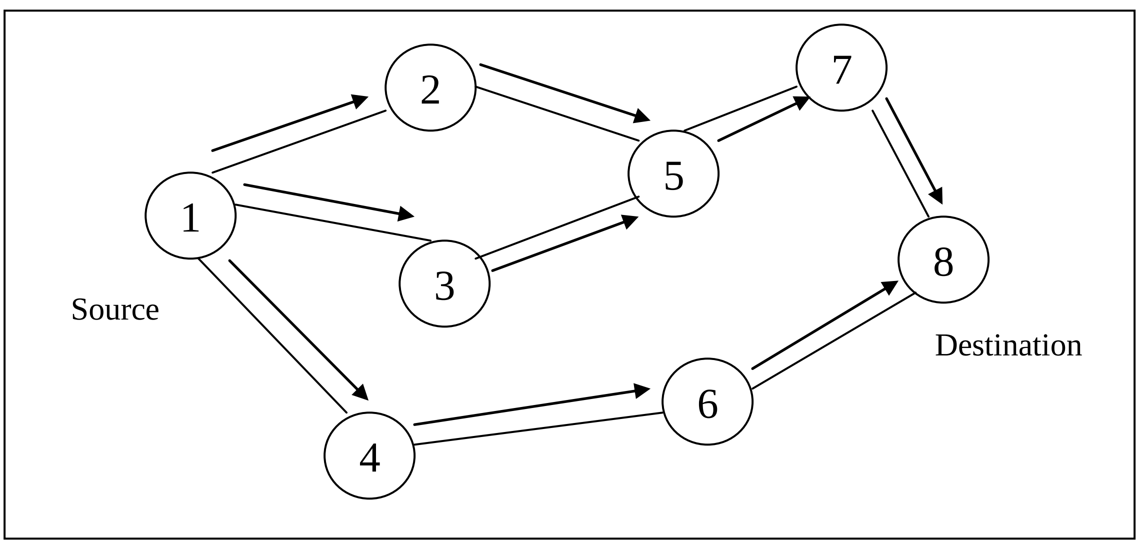

Figure 1.3 Route discovery process of AODV [3]

Once the RREQ reaches the destination which is node 8, the destination node responds a route reply (RREP) packet back to the source node with the best possible route as shown in Figure 1.4. Hence, all the nodes participating at route discovery process will have the ability to update their routing tables accordingly. Figure 1.4 shows the route reply process from destination node, 8.

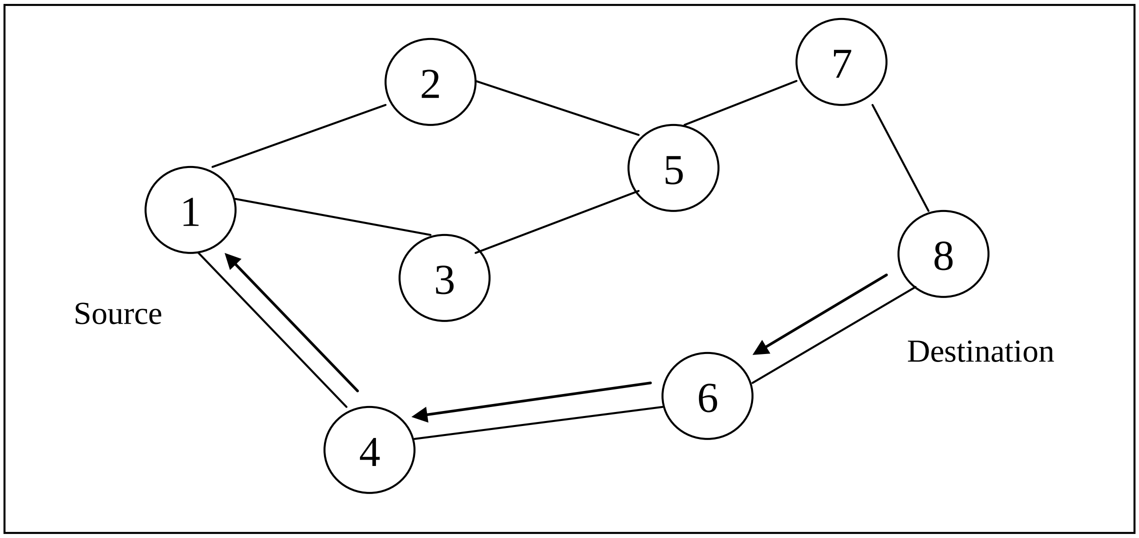

Figure 1.4 Route reply process of AODV [3]

2.1.4 DSR

The Dynamic Source Routing (DSR) Protocol is a source-routed on-demand routing protocol. A node maintains route caches containing the source routes that it is aware of. The node updates entries in the route cache as and when it learns about new routes [3]

The two major phases of the protocol are: route discovery and route maintenance. When the source node wants to send a packet to a destination, it looks up its route cache to determine if it already contains a route to the destination. If it finds that an unexpired route to the destination exists, then it uses this route to send the packet. But if the node does not have such a route, then it initiates the route discovery process by broadcasting a route request packet. In Figure 1.5, it is seen the source node, 1 is broadcasting its message to all the nearest neighbors. This is called the route discovery process. The route request packet contains the address of the source and the destination, and a unique identification number. [3]

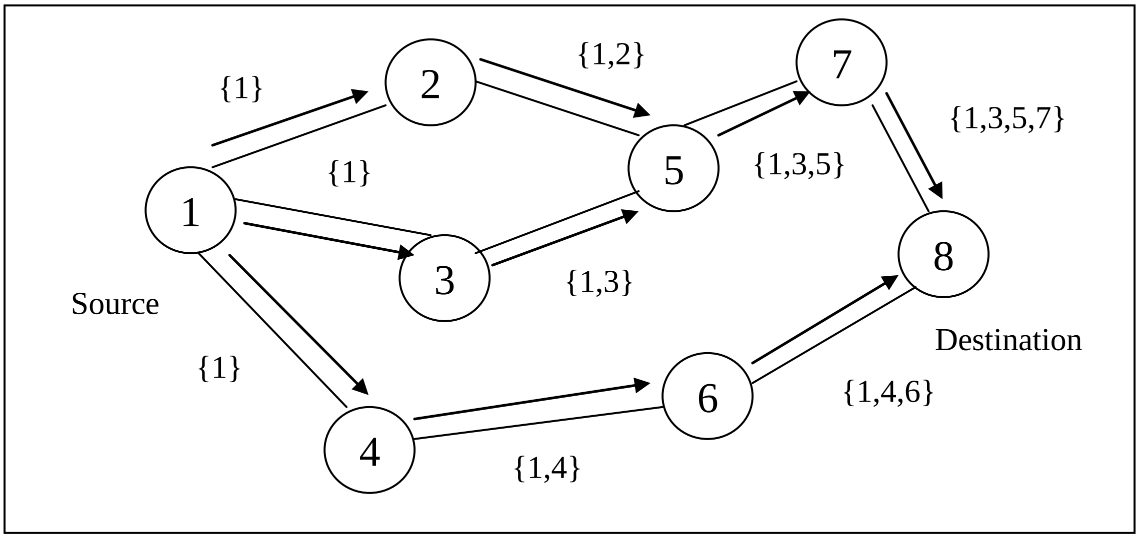

Figure 1.5 Route discovery process of DSR [3]

Each intermediate node checks whether it knows of a route to the destination. If it does not, it appends its address to the route record of the packet and forwards the packet to its neighbors. From figure 1.5, it is seen that the rest of nodes 2 to 7 has appended their addresses to the route record. To limit the number of route requests propagated, a node processes the route request packet only if it has not already seen the packet and it's address is not present in the route record of the packet. As the route request packet propagates through the network, the route record is formed as shown in figure 1.5.

A route reply is generated when either the destination or an intermediate node with current information about the destination receives the route request packet. A route request packet reaching such a node already contains, in its route record, the sequence of hops taken from the source to this node.[3]

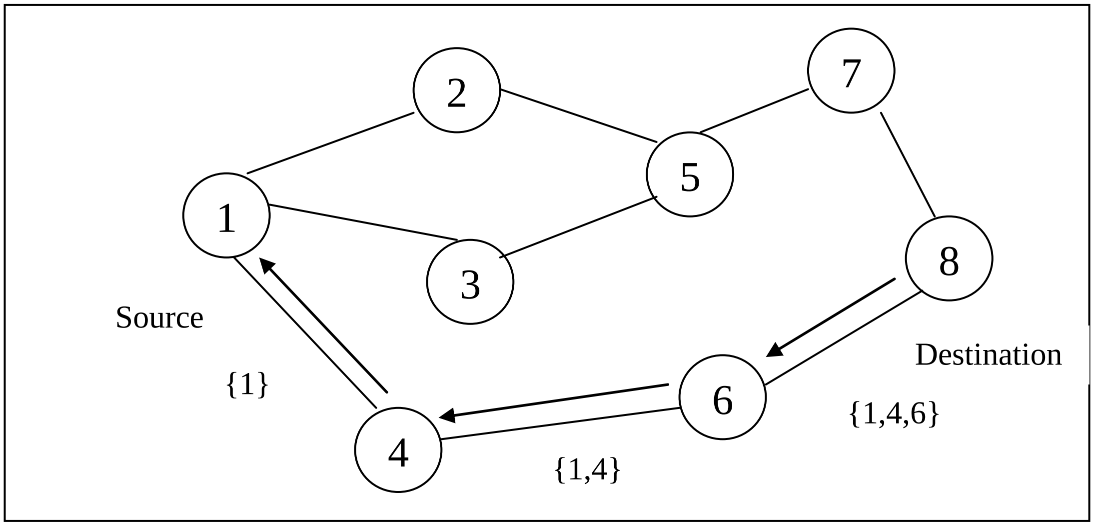

Figure 1.6 Route Reply process of DSR [3]

If the route reply is generated by the destination then it places the route record from route request packet into the route reply packet. This can be seen in Figure 1.6. On the other hand, if the node generating the route reply is an intermediate node then it appends its cached route to destination to the route record of route request packet and puts that into the route reply packet. Figure 1.6 shows the route reply packet being sent by the destination itself. To send the route reply packet, the responding node must have a route to the source. If it has a route to the source in its route cache, it can use that route.

The reverse of route record can be used if symmetric links are supported. In case symmetric links are not supported, the node can initiate route discovery to source and piggyback the route reply on this new route request.

DSRP uses two types of packets for route maintenance:- Route Error packet and Acknowledgements. When a node encounters a fatal transmission problem at its data link layer, it generates a Route Error packet. When a node receives a route error packet, it removes the hop in error from its route cache. All routes that contain the hop in error are truncated at that point. Acknowledgment packets are used to verify the correct operation of the route links. This also includes passive acknowledgments in which a node hears the next hop forwarding the packet along the route. Table 2.1 shows the overall characteristics of the mobile ad-hoc routing protocols as shown in [7].

Protocol Property	**DSDV**	**DSR**	**AODV**	**OLSR**
Multicast routes	No	Yes	No	Yes
Distributed	Yes	Yes	Yes	Yes
Unidirectional Link Support	No	Yes	Yes	Yes
Multicast	No	No	Yes	Yes
Periodic broadcast	Yes	No	Yes	Yes
QoS Support	No	No	No	Yes
Routes maintained in	Route table	Route cache	Route table	Route table
Reactive	No	Yes	Yes	No

Table 1.1 Characteristics of the MANET routing protocols [7]

1.5 Summary

In this chapter, a basic understanding of mobile ad hoc networks (MANET) is provided. Definitions of mobile ad-hoc networks (MANET)s and its applications are discussed. The concept of TCP, its performance overview in wireless networks and the importance of routing protocols in multi-hop network towards TCP performance is also discussed.

In this chapter, the concept of routing, its types and its uses in mobile ad-hoc networks are discussed. There are two major types of ad-hoc routing protocols - proactive and reactive. DSDV and OLSR are proactive routing protocols where routing is table-driven and it is difficult to adapt to changing environment. On the other hand, AODV and DSR are reactive routing protocols, more suitable for changing environment since they are not table-driven.

Chapter 2. Background and Related Work

2.1 Background

Many researchers in the couple of decades evaluated the performance of the various MANET routing protocols and made different conclusions. However, the behavior of these routing protocols can be tested to its limit only if a wide variety of parameters are considered over a wide scale of networks. Parameters like normalized routing load and packet delivery ratio fluctuates a lot with change in the load of the network. The behavior of these routing protocols needs to be analyzed at varying network load, network size and node density in order identify the most adaptive and efficient routing protocol.

Most of the work done in the past makes a comparison of the important parameters like normalized routing load, average end to end delay and packet delivery ratio. However, most of them keeps the comparison to two or three protocols. As a comprehensive approach to study the performance of these routing protocols, we have considered all the proactive and reactive routing protocols into account viz. DSDV, OLSR, AODV and DSR. A performance comparison is drawn in terms of the parameters - average throughput, packet delivery fraction and number of packets dropped. Additionally, a packet analysis is also done which includes analysis between the number of packets sent, received and forwarded by different routing protocols in different scenarios.

In the next section, we give an overview of the work done in this field and make a head to head comparison to our results.

2.2 Related work and comparison

In this section, previous work on the performance analysis of the mobile ad-hoc routing protocols is overviewed. It is observed that some papers consider less than four protocols in their comparison while some others do not take mobility into account. Some papers vary mobility but do not consider speed as an important variable. Network load or varying the number of nodes is a big factor which impacts the routing performance of the ad-hoc protocols but very few papers have made a comprehensive analysis over it. For instance, [5] makes a performance comparison of the routing protocols for ad hoc networks with a fixed number of nodes. They compare the standard Djikstra algorithm with OLSR, AODV and DSR. According to them, with CBR sources, the performance of packets correctly delivered is quite high (over 90%). Although this is true, however, increase in traffic load significantly decreases the overall packet delivery ratio. This is simulated and studied in our research.

When the mobility model is considered, Random way-point model is the optimum model for dealing with MANET networks since the nodes can move in any direction. [5] gives a performance evaluation comparing three routing protocols - AODV, DSR and OLSR. This paper essentially discusses multimedia transmission over 50 nodes and analyzes the performance of the routing protocols. Unlike our simulation scenario which gives an analysis over a wide range of nodes, this paper [5] has a constant number of nodes and follows a Manhattan Grid model which opposes the random way-point model considered in this research. Performance parameters like packet delivery ratio, end to end delay and routing overhead have been analyzed but mostly these parameters are plotted against number of connections which is considered as the main variable by [5]. According to [10], pause time basically determines the mobility rate of the model because as pause time increase, mobility increases. Thus pause time is considered as one of the most important variable for analyzing mobility rate in this research.

A very good comparison between DSDV, DSR and AODV is made in [6] and the network load is increased by 5 nodes for every simulation. [6] compares only three protocols unlike this research which compares all the four. It has considered a maximum of 20 nodes unlike this research which compares the network analysis up to 70 nodes. Additionally, [6] does not clearly state how many sources and receivers are communicating and what type of TCP agent is used. It is also not clear what is the maximum speed of the nodes when analysis of the routing protocols are made.

Only two ad-hoc routing protocols DSR and AODV are analyzed in [7] compared to four protocols which has been done in this research. They mostly studied the inter-layer interactions between the physical and the MAC layer and their performance implications. Most of the simulation parameters and performance parameters were similar compared to this research. They have also included the Random Waypoint Model as the backbone model, run for 50 and 100 nodes with different simulation time for each metric. Speed was varied between 0-20 m/s and number of sources between 10-30. They have found out that DSR demonstrated significantly lower routing load than AODV and this is comparable to the results of this research. The paper observes that AODV outperforms DSR in terms of packet delivery ratio even when the network load is increased which is comparable to the results that drawn by this research. However, this paper strictly adheres to comparison of only two ad-hoc routing protocols and gives no comparison with proactive routing algorithms like OLSR and DSDV.

A performance comparison between all the four MANET routing protocols AODV, DSDV, DSR and OLSR is given in [8]. However this paper is based on theoretical analysis and it does not provide anything related to simulation and result analysis. The paper does not speak anything specific about performance analysis related to the OLSR protocol.

In [11], the performance parameters analyzed are the mobility rate, network load and network size. DSDV, AODV, DSR and TORA are analyzed in this paper but OLSR is not compared. Only fixed number of sources have been taken in this paper unlike our research which also takes in to account different network load scenarios. This paper gives a reason why CBR sources are preferred over TCP source and why pause time is considered an important factor for mobility. Thus, pause time is considered as one of the major variable in comparing the performance analysis of the routing protocols in this research.

A comparative investigation on the performance of the routing protocols DSR, AODV and DSDV is done in [13]. This paper includes mobility and speed into account which is a major contribution. In order to verify the results produced by the paper, this paper is simulated and the results are successfully replicated by this thesis. They have studied the effects of varying node mobility rate, scalability and maximum speed on the performance of DSR, AODV and DSDV. Their simulations indicate that reactive routing protocols AODV and DSR perform better than proactive routing protocol DSDV. The paper claims that DSR produces large overhead with respect to network size and hence it is less scalable. However, since the paper strictly adheres to a fixed number of nodes, this claim is not proven. In this thesis, another proactive routing protocol, OLSR is simulated and contrasting results to [13] are found. OLSR is found to be better than DSDV in terms of proactive routing. Additionally in this thesis, different network sizes are implemented and a comprehensive performance evaluation of all the routing protocols is carried out, which is not done in [13].

2.3 Summary

In this chapter, an overview of the background of this research area is discussed. Related work to this research is presented in this chapter. Similarities and contrasting features with important conference and journal papers are also discussed. Some of the important considerations for a comprehensive performance that were not considered by some previous papers are pointed out.

Chapter 3. Proposed Model and Performance Parameters

3.1 Proposed Model

A model is proposed based on some of the important assumptions that are not considered by the papers mentioned in the previous section. The entire research work is divided into three phases - mobility, speed and network load to make a comprehensive performance analysis of the mobile ad-hoc routing protocols. Three major performance parameters - average throughput of the network, packet delivery fraction and number of packets dropped are considered to determine the performance of the ad-hoc routing protocols.

In the first and second phase, [13] is simulated to verify accuracy of the protocols. However, OLSR is added to analyze its performance against the other protocols. In the first phase, mobility is considered keeping the number of nodes and the CBR sources constant. In the second phase, maximum speed is the main variable and the performance parameters are plotted against speed. The third phase deals with the network density to determine how the ad-hoc routing protocols perform against various network loads. CBR sources are roughly taken as one-third of the number of nodes to maintain consistency. This model determines the performance analysis of the routing protocol in a varied scale of networks- it includes small, medium and large networks.

Pause time is considered as one of the main variable for analysis since maximum mobility variance is one of the most important factor in moving nodes [10]. A complete packet analysis which gives the number of packets sent, received, dropped and forwarded is analyzed to check the performance of the mobile ad-hoc routing protocols under different traffic conditions.

The figures below show the scenarios with three different scales of network. We have assumed that anything below 25 nodes is a small network, 25- 50 is medium and 50-100 nodes are a large network.

a. Small Network Scenario

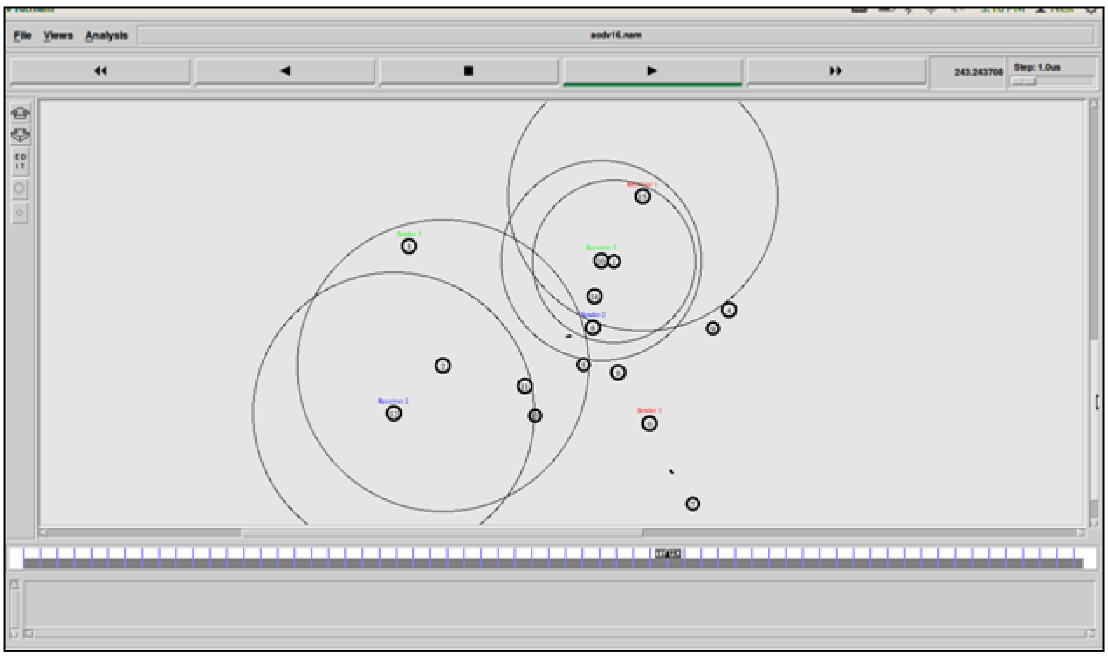

Figure 3.1 A small network scenario consisting of 16 nodes

In a small network scenario as shown in Figure 3.1, the number of nodes forming the network is usually less. There are 16 nodes in Figure 3.1 and the nodes are randomly moving in different directions. The circles around the nodes are wireless transmission range and based on the range of the different mobile nodes, the data is transmitted from the source to the destination.

In a medium network scenario as shown in Figure 3.2, the number of nodes making the network are 35. They are larger than the small network scenario and different routing

protocols behave in different ways owing to the change of the network conditions.

b. Medium Network Scenario

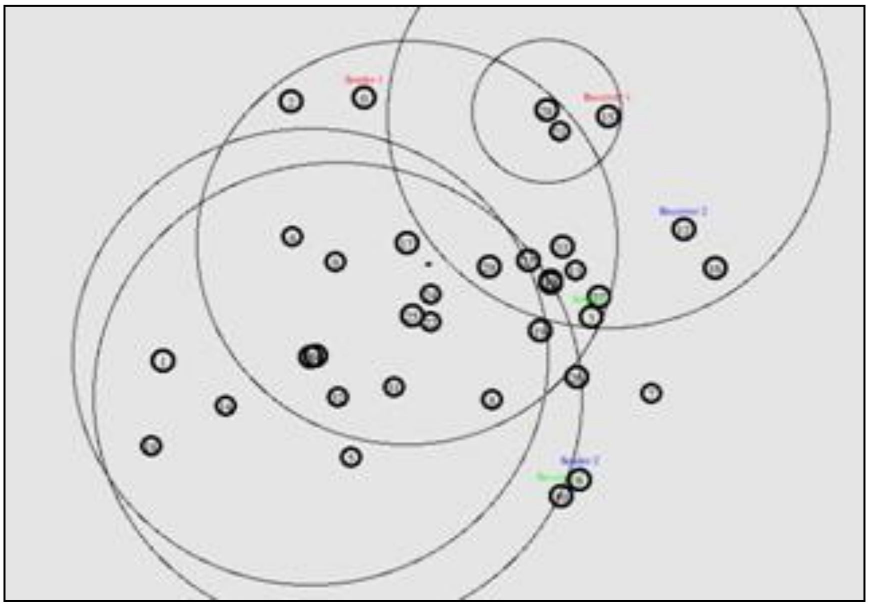

Figure 3.2 A medium network scenario consisting of 35 nodes

Figure 3.3 shows a large network scenario which consists of 70 nodes. In such large networks, it is difficult to achieve 100% packet delivery fraction. This is verified in the forthcoming sections of this research.

The complete simulation setup and the parameter consideration has been detailed in section 5.1

c. Large Network Scenario

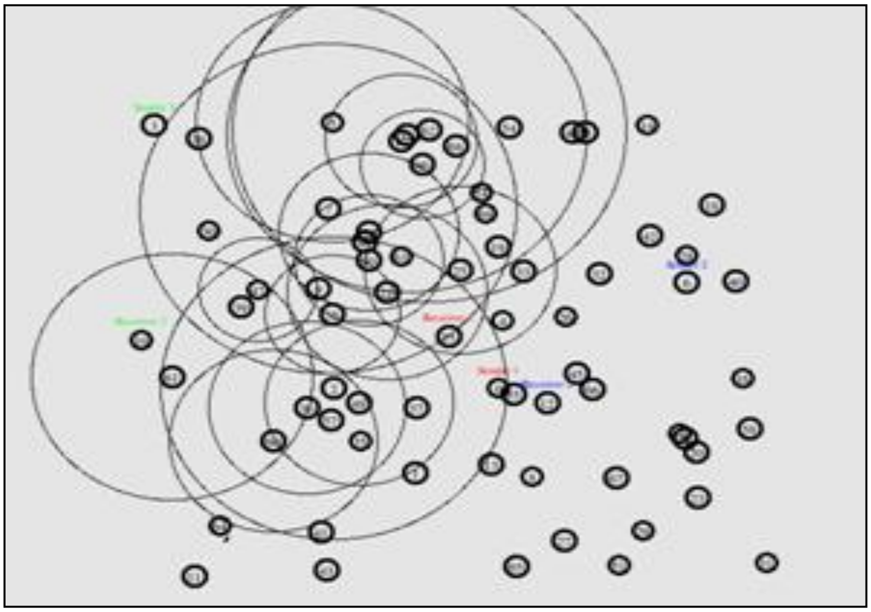

Figure 3.3 A large network scenario consisting of 70 nodes

3.2 Performance Parameters

There are three main performance parameters that are considered in this research - Average throughput, packet delivery fraction and the number of packets dropped. Average Throughput determines the stability of the network in different traffic conditions. Packet delivery fraction accounts to the percentage of packets delivered when the network is subjected to different traffic conditions. Number of packets dropped is considered to observe if the number of packets received is affected more by forwarded packets or dropped packets. These three parameters are evaluated through the three phases of the research to make the performance analysis of the ad-hoc routing protocols.

3.2.1 Throughput

It gives the fraction of the channel capacity used for useful transmission (Data packets correctly delivered to the destination) and is defined as the total number of packets received by the destination. It is in fact a measure of the effectiveness of a routing protocol measured in bits/second.

Throughput = (Number of packets sent * 8 * 512) / Simulation Time (3.1)

3.2.2: Packet Delivery Fraction/Ratio

It is the ratio of data packets received to packets sent. It tells us about the fraction of the packets delivered from source to destination when the network is subjected to different traffic conditions. It also gives an idea about the number of packets dropped or forwarded by the routing protocol.

Packet Delivery Fraction = Number of packets Received/ Number of packets sent (3.2)

3.2.3: Pause Time

The parameter which is of primary importance is pause time. Pause time basically determines the mobility rate of the model, as pause time increases the mobility rate decreases[10]. Pause time is the amount of time taken by each of the moving nodes before they start transmitting packets. When the pause time is high, the wait time for the nodes is high and the mobility is low because the nodes are not continuously sending packets. When the pause time is low, the wait time for the nodes is low and hence the mobility is high. It means the nodes are constantly sending packets without any wait time.

3.3 Summary

In this chapter, the proposed model of this research is discussed. The research is divided into three phases - mobility, speed and network load to make a comprehensive performance analysis of the mobile ad-hoc routing protocols. The performance parameters that are considered in this research are average throughput, packet delivery fraction and the number of packets dropped. These performance parameters are plotted against pause time, maximum speed and number of nodes to make a comprehensive analysis.

Chapter 4. Simulation Results

4.1 NS 2.35 and patches

NS is a discrete event simulator targeted at networking research. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. NS is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. NS uses two languages because simulator has two different kinds of things it to do. On one hand, detailed simulations of protocols requires a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets.[42]

NS 2.35 has been used in this research. It is open source software and hence can be found on the internet for free. NS 2.35 can be installed over a Windows or a Linux operation system. Ubuntu 13.10 has been used as the operating system. NS 2.35 has been installed on the Ubuntu and simulations have been carried out.

TCL scripts are written to simulate various network scenarios and they are executed to get a trace(trc) file and a network animator (nam) file. The trace file contains the details of the entire simulation details like different time slots, communication between the different nodes, packet size and name details, source and destination address details, MAC addresses and various other details of the environment. In order to calculate different network parameters, data is fetched from this trace file via awk script and plotted into a graph. Various awk scripts are written to fetch the data from these trace files and calculate the required network parameters. Network animator (nam) files are produced once the tcl script is executed. These files contain details about the animation of the nodes. The network simulator contains an animator output or a graphical user interface (GUI) where the details of the node scenarios and movement information is fetched from the tcl script and presented in front of the user. Once the nam file is executed, the network animator shows up and accurately displays the scenarios created by the user. Once the play button is activated, the network animator shows the data flow between the nodes along with the elapsed time.[42]

UM-OLSR patch has been applied to NS 2.35. This patch is applied for the implementation of the OLSR protocol. NS does not have the OLSR protocol in-built to perform simulations and hence the patches had to be applied. UM-OLSR complies with IETF RFC 3626 and supports all functionalities of OLSR plus the link-layer feedback option. After the patch was applied, the NS 2.35 code was configured, builds and tested for conducting successful simulations.

4.2 Simulation setup

The simulation environment is set up in a way that four major protocols can be analyzed with different parameters. From a reference point of view, a paper [13] which performs investigations on three routing protocols - AODV, DSR and DSDV is simulated to verify consistency. The results produced by [13] are successfully replicated. Over 160 simulation scenarios are considered to get the results of this research.

Simulations are divided into three phases- mobility, speed and network load. In the first phase and second phase, the number of nodes are kept constant like [13] and average throughput, packet delivery fraction and packet loss are calculated to replicate the results produced by [13]. In the last phase, the same parameters are calculated by varying the number of nodes.

The first and the second simulation environment consists of 50 nodes forming an ad-hoc network, moving over a 670 meter by 670 meter flat space for 200 seconds of simulation time. NAM animation tool is used for viewing network simulation networks and real world packet trace data. The number of CBR sources used in these two simulation environments is kept as 10 in order to replicate the results of [13]. In the third simulation environment, the scalability of the networks is measured by varying the load of the network. The number of nodes is used as a variable and the performance of all the four ad-hoc routing protocols - AODV, DSR, DSDV and OLSR are carried out. In this case, every time the number of nodes is increased, the CBR sources are set to one-third of the number of nodes to follow a consistent pattern of traffic.

Each run of the simulator accepts two kinds of inputs- the movement file that describes the movement of each node, and a connection pattern file which sets up random traffic generated by the type of traffic connection (CBR in our case). In NS, the movement file has all the movements of the nodes at different times with different speeds. This file is generated by a *setdest* command. The connection pattern file determines the type of traffic connection if it is a TCP or CBR connections between the nodes. It also gives an idea about the number of sources and the total number of connections made by the nodes in that simulation time. This file is generated by a *cbrgen* command. In order to enable direct and fair comparisons, protocols are simulated under identical loads and environmental conditions. The different scenario files are pre-generated with varying movement patterns and traffic loads.

In the next three sections, a detailed description of the simulation parameters used and results obtained from the simulation is provided. The graphs are analyzed and rational conclusions are drawn to support which protocol is best suited for what kind of scenarios in an ad-hoc network.

4.3 Performance analysis by varying mobility

Parameters	Value
Simulation Time	200 seconds
Environment Size	670 x 670
Packet Size	512 bytes
Traffic type	CBR
Packet rate	4 packets/second
Mobility model	Random Way-point model
CBR sources	10
Maximum Speed	20 m/s
Pause Time	0, 50, 100, 150, 200
Protocols	AODV, DSR, DSDV, OLSR
Number of nodes	50

Table 4.1 Simulation setup phase 1

Table 4.1 shows the simulation parameters used in phase 1 of the research. In this simulation, the number of nodes is kept constant at 50 and the pause time or the mobility of the nodes is varied. The results in Figure 4.1 obtained are consistent with [13] but it is observed that the added protocol OLSR dominates its peer, DSDV throughout the simulation. OLSR performs better than DSDV in this case and stands almost as tall as the reactive protocols, AODV and DSR. It can concluded that DSDV does not have a significantly higher throughput when mobility is high i.e. Pause time is 0 but as mobility decreases, performance of DSDV gets better. This is because DSDV has difficulty finding routes in higher mobility because of its proactive nature. However, reactive protocols AODV and DSR maintains consistency in varying mobility and performs better than the proactive protocols OLSR and DSDV.

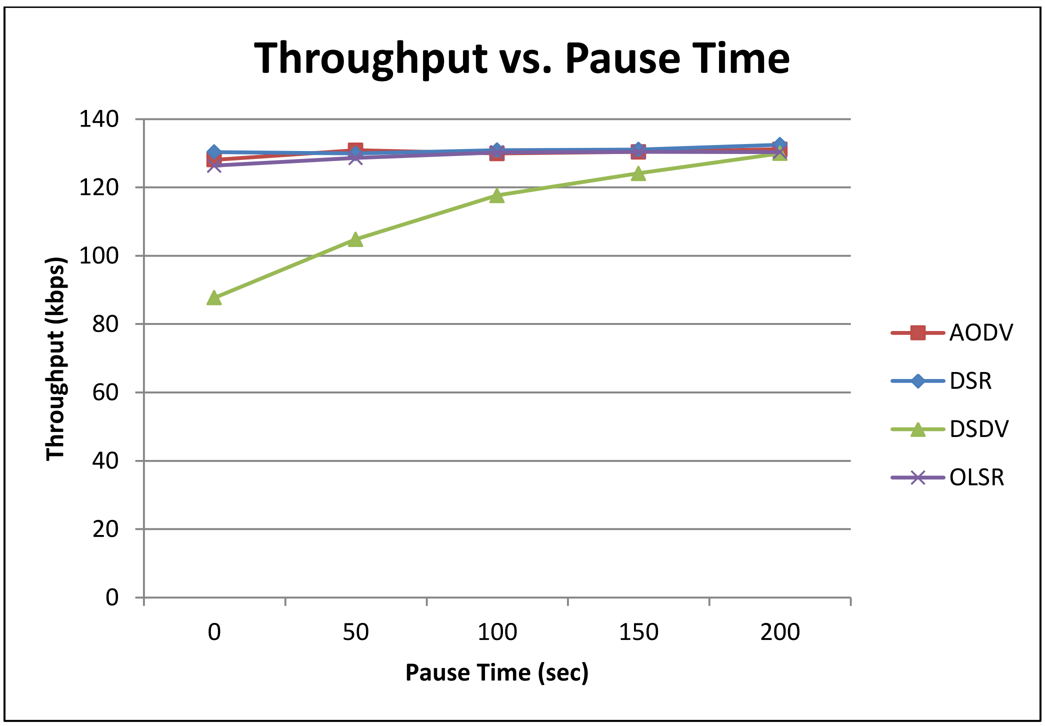

Figure 4.1 Throughput plot by varying mobility

Next, the packet delivery fraction is analyzed by varying mobility. All the protocols deliver a greater performance of packet delivery fraction except DSDV. From Figure 4.2, it is observed that at higher mobility, performance of DSDV drops down to as low as 70%. OLSR proves better performance compared to DSDV as its performance is slightly below AODV and DSR's packet delivery fraction.

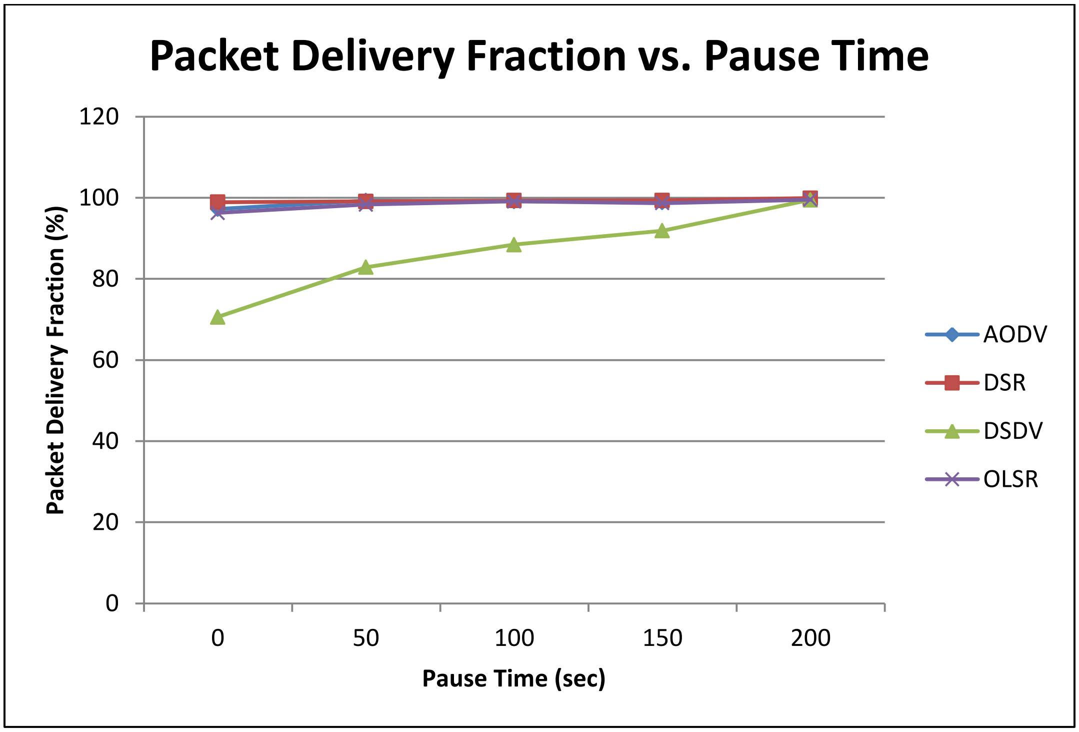

Figure 4.2 Packet Delivery Fraction plot by varying mobility

Figure 4.3 shows the number of packets dropped when mobility is varied. DSR shows the best performance out of all the four protocols because only 97 packets are dropped at maximum mobility. Performance of AODV and OLSR are comparable. DSDV showed worst performance as the number of dropped packets was close to 2514 at maximum mobility. As the mobility is decreased or the pause time is increased, DSDV performs well because the nodes get enough time to update the routing tables.

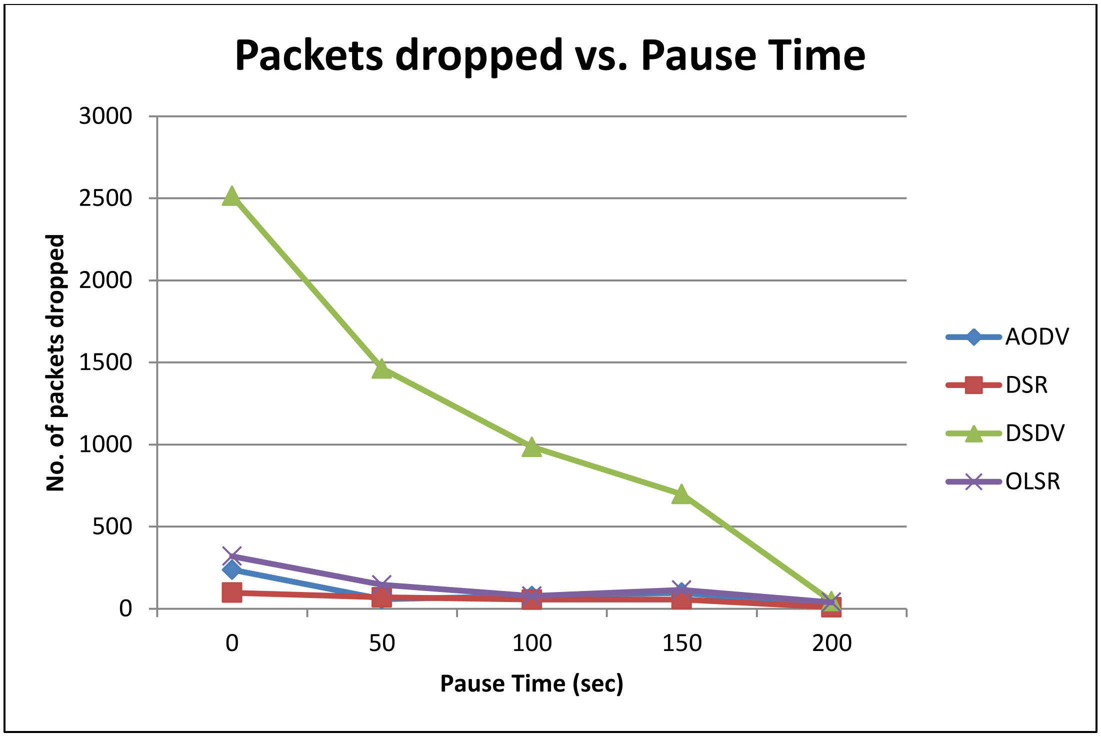

Figure 4.3 Plot of the packets dropped by varying mobility

4.4 Performance analysis by varying speed

Table 4.2 shows the simulation parameters used in phase 2 of this research. In this simulation, the number of nodes is kept constant at 50 and the maximum speed of the nodes is varied. The pause time is taken as 0 seconds so that maximum mobility variance can be considered. In this phase, throughput, packet delivery fraction and the number of packets dropped are calculated to replicate results of [13]. Results of [13] are successfully replicated by varying speed and its performance is compared to another proactive protocol, OLSR. The maximum speed has been varied from 1 meters/sec (3.6 km/hr) that corresponds to walking at a slow speed to 50 meters/sec (180 km/hr), the speed of a very fast car.

Parameters	Value
Simulation Time	200 seconds
Environment Size	670 x 670
Packet Size	512 bytes
Traffic type	CBR
Packet rate	4 packets/second
Mobility model	Random Way-point model
CBR sources	10
Maximum Speed	1, 2, 5, 10, 20, 50 m/s
Pause Time	0
Protocols	AODV, DSR, DSDV, OLSR
Number of nodes	50

Table 4.2 Simulation setup phase 2

The results in Figure 4.4 obtained are consistent with [13] but it is observed that the added protocol OLSR dominates its peer, DSDV throughout the simulation. DSR shows maximum and consistent throughput throughout all speeds. It has an average speed of 131 kbps which is higher than AODV, 129 kbps and OLSR, 125 kbps. DSDV suffers decrease in throughput close to 68 kbps at maximum speed(5 meters/sec). This is because of frequent link changes and connection failures. It can also be observed throughput for OLSR has started decreasing at high speed because of its proactive nature.

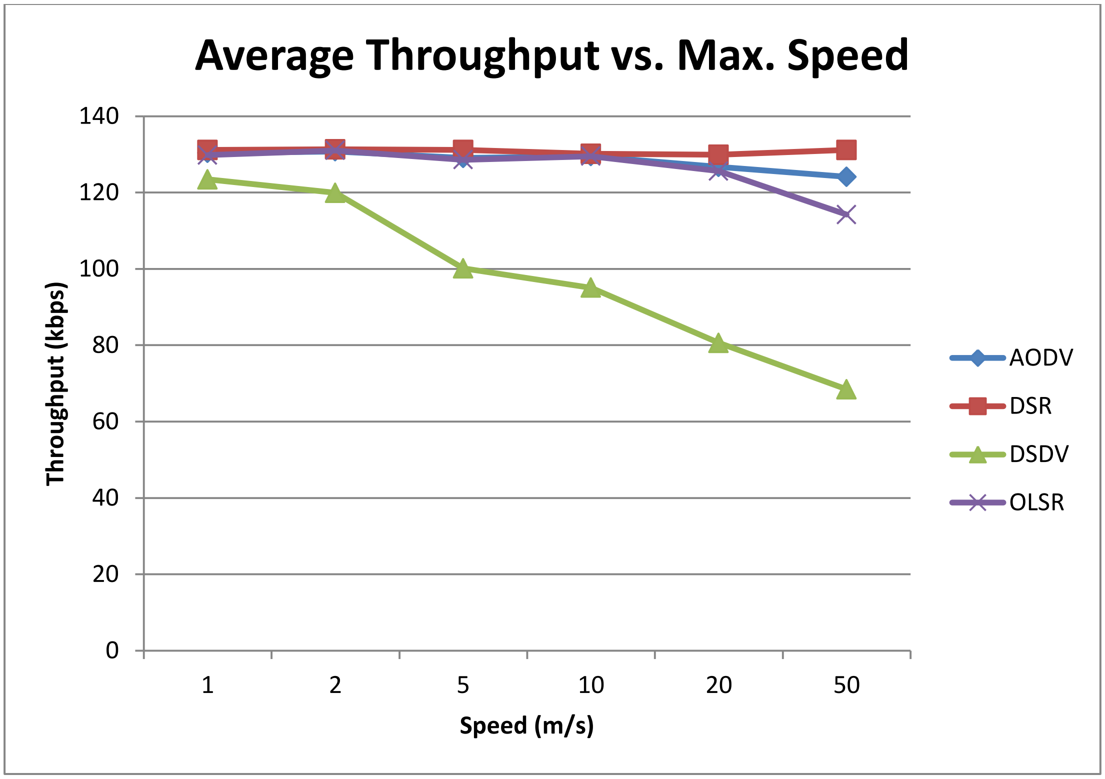

Figure 4.4 Throughput plot by varying speed

When the packet delivery fraction is calculated against varying speed, it is observed from Figure 4.5 that DSR again outperforms all the protocols at all speeds maintaining a packet delivery fraction close to 100%. AODV's performance is comparable to DSR delivering almost 98% of the packets. DSDV delivers close to 96% of the packets at low speed but could not keep the same rate with the increase in speed because of its frequent link changes and connection failures. Packet delivery in DSDV drops to as low as 51% in high speed. OLSR again performed better than DSDV as its performance closely matched with the reactive protocol delivering close to 98% of the packets.

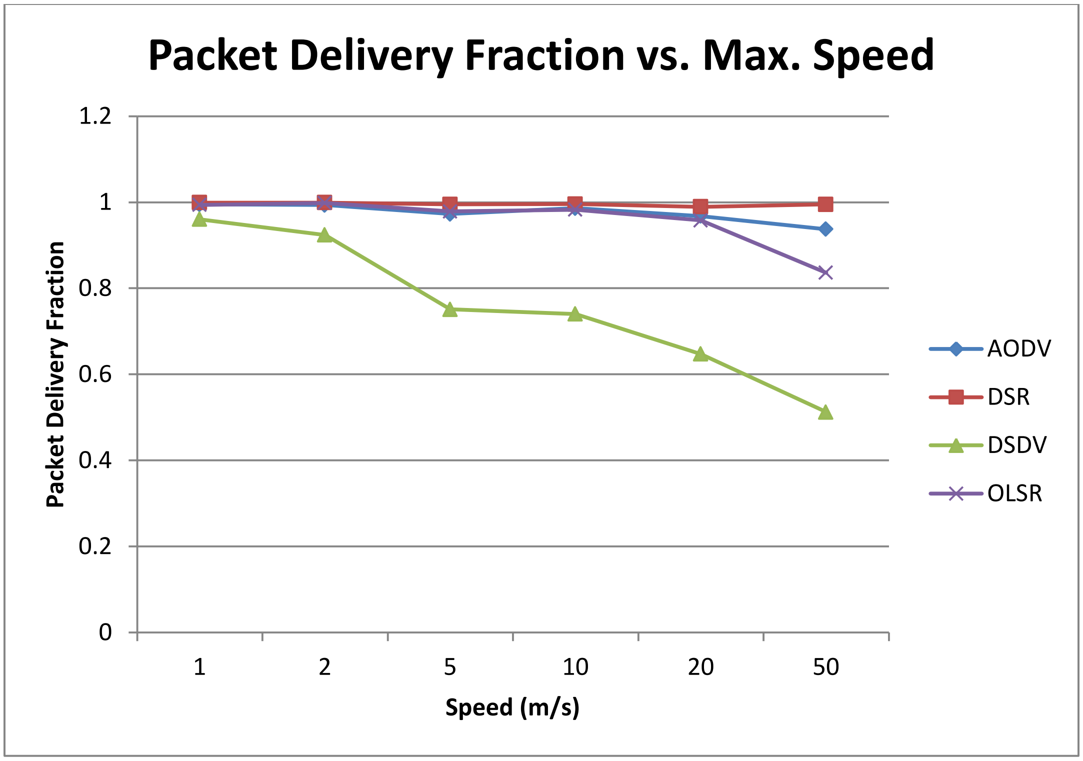

Figure 4.5 Packet Delivery Fraction plot by varying speed

In Figure 4.6, the number of packet dropped is plotted by varying speed. DSR once again shows optimum results with the number of dropped packets being significantly low. Even high speed, DSR is able to maintain a low drop rate because of its efficiency in its dynamic routing algorithm. AODV and OLSR performed well in low speed but as the speed increased, the number of dropped packets also increased. DSDV once again failed to perform well in high speed as the number of dropped soared well above 4000 at a maximum speed of 50 meters/second.

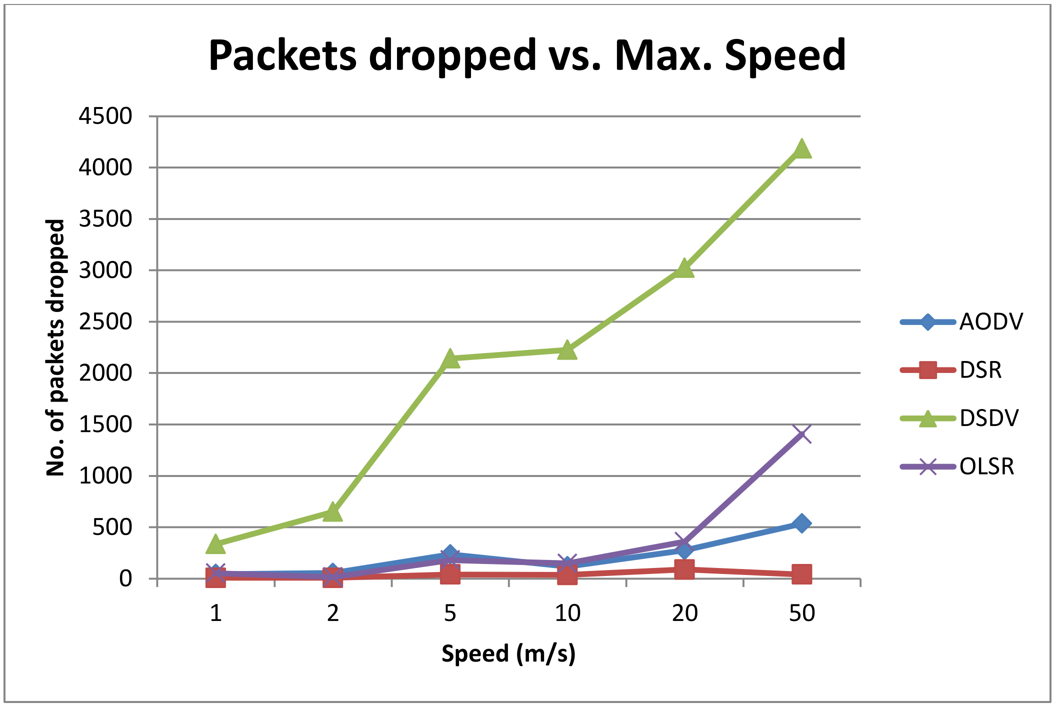

Figure 4.6 Plot of packets dropped by varying speed

4.5 Performance analysis by varying network load

This is the third phase of the simulation environment where performance of the routing protocols is evaluated by varying the network load. In this phase, the same performance parameters- throughput, packet delivery ratio and packets dropped are analyzed by changing the load in the network. This phase is required to measure the scalability of the routing protocols in small, medium and large networks. As such, the number of nodes has been varied from 20 nodes to 100 nodes so that a small, medium and a large network can be simulated. Since the number of nodes is varied, the number of CBR sources is also changed. In the previous phases, the number of CBR sources was 10 which were consistent with the number of nodes, 50. But in this phase, the number of CBR sources is roughly taken as one-third of the total number of nodes to maintain consistency in traffic. The speed is kept at 20 meters/second and pause time at 0 seconds to simulate maximum mobility variance.

Parameters	**Value**
Simulation Time	200 seconds
Environment Size	670 x 670
Packet Size	512 bytes
Traffic type	CBR
Packet rate	4 packets/second
Mobility model	Random Way-point model
CBR sources	One-third of the nodes
Maximum Speed	20 m/s
Pause Time	0
Protocols	AODV, DSR, DSDV, OLSR
Number of nodes	20, 40, 60, 80, 100

Table 4.3 Simulation setup phase 3

Figure 4.7 shows the average throughput of the ad-hoc routing protocols under varying network load. It is seen that AODV performs the best compared to the other protocols with a peak throughput of 167.5 kbps. DSR could not sustain the performance at higher network load. DSDV significantly has lower performance because of frequent link changes and connection failures. OLSR performs better than DSR and DSDV which makes it capable of running in large networks but may result in heavy overload and congestion problems according to [16]. OLSR requires extra time to set up routing tables before delivering packets but packets seems to be thoroughly forwarded and received which gives a consistently high average throughput.

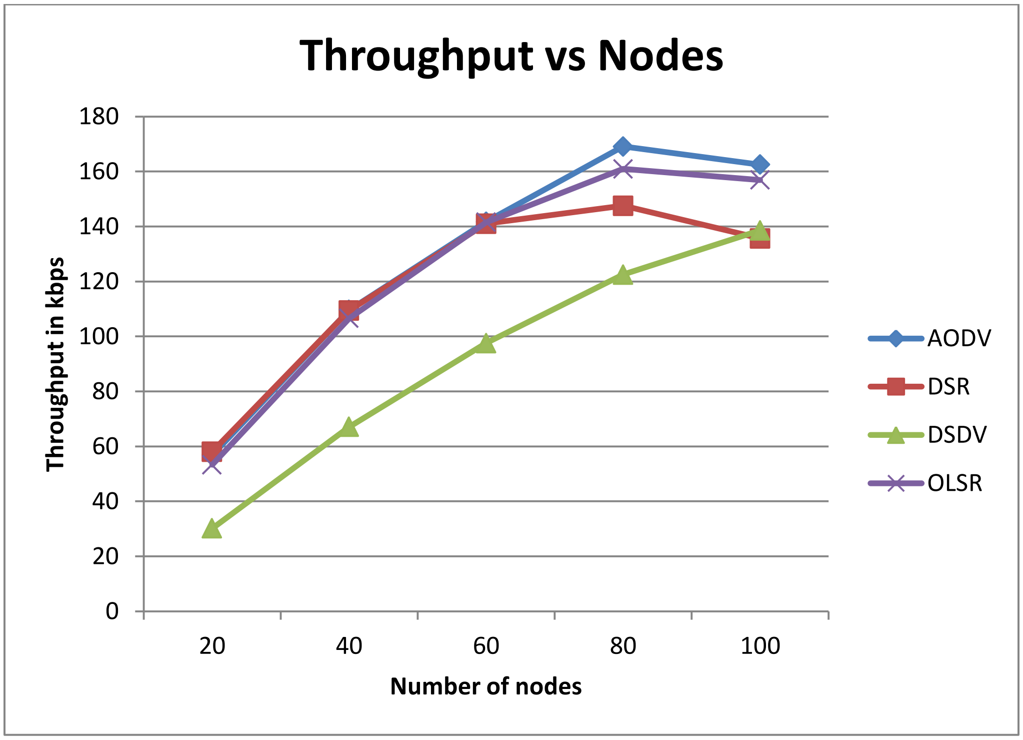

Figure 4.7 Throughput plot by varying network load

Figure 4.8 gives the packet delivery fraction of all the protocols when the nodes are varied. Looking at the trend, it can observed when the network load is increased, the packet delivery fraction for all the protocols gets reduced. DSR has a peak packet delivery fraction of close to 100% when it is a small network i.e. number of nodes is 20. But as the load is increased, the performance degrades. For a large network scenario (100 nodes), packet delivery comes down to as low as 39% which shows that DSR does not perform well when the network size is complex. AODV and OLSR shows similar trend with AODV slightly showing better performance in small and large networks. DSDV has a low packet delivery fraction throughout the different scale of networks but it has a certain kind of consistency. The packet delivery fraction of DSDV is consistent between 55% to 67% throughout the different scale of networks.

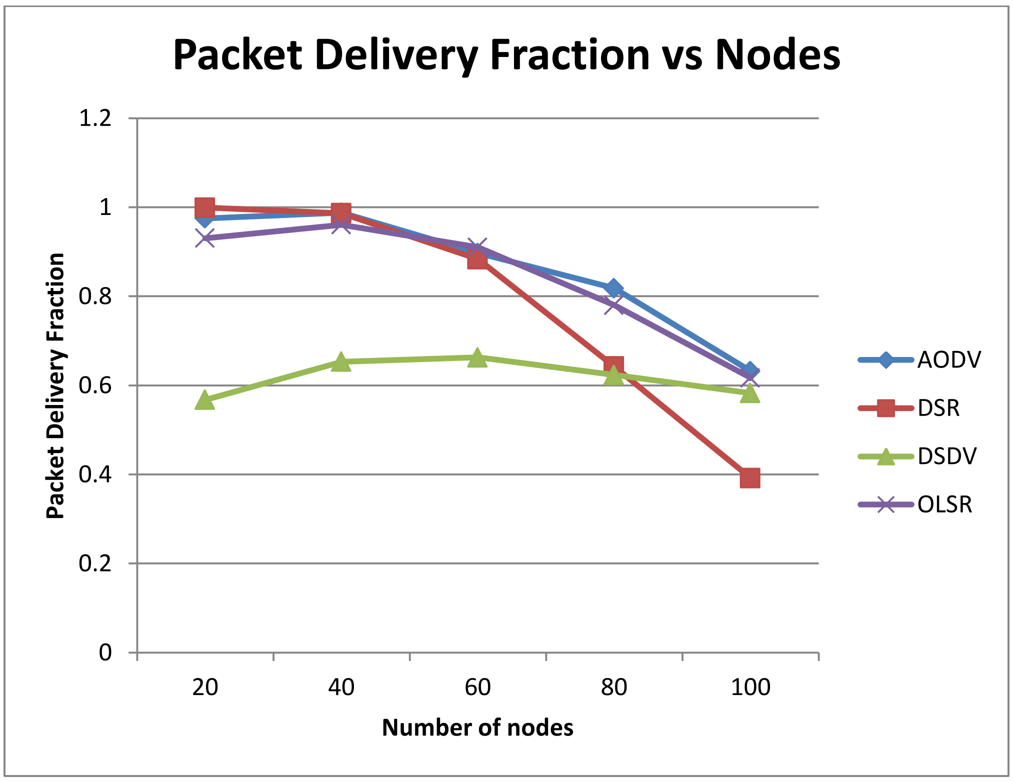

Figure 4.8 Packet Delivery Fraction plot by varying network load

In Figure 4.9, it is observed that all the protocols are vulnerable to large networks as more number of packets started dropping after 60 nodes. DSDV started dropping packets as many as 1542 even in small scale networks and more than 6700 packets in large scale networks. This makes DSDV a tougher choice for an efficient routing protocol for loaded ad-hoc networks. DSR performs better in small network but loses as many as 9800 packets when the size of the network is increased. DSR's performance is comparable to its other peer, AODV but it fails to keep up the performance when the load on the network is increased. This is probably because as the network size increases, DSR becomes more aggressive with caching. In large networks, routes become larger thus increasing the probability of route errors and stale routes which in turn is enough to drop more packets. OLSR's performance is comparable to AODV but it lost more packets than AODV in small as well as large networks. At large networks, when AODV lost close to 5900 packets while OLSR lost close to 6854 packets.

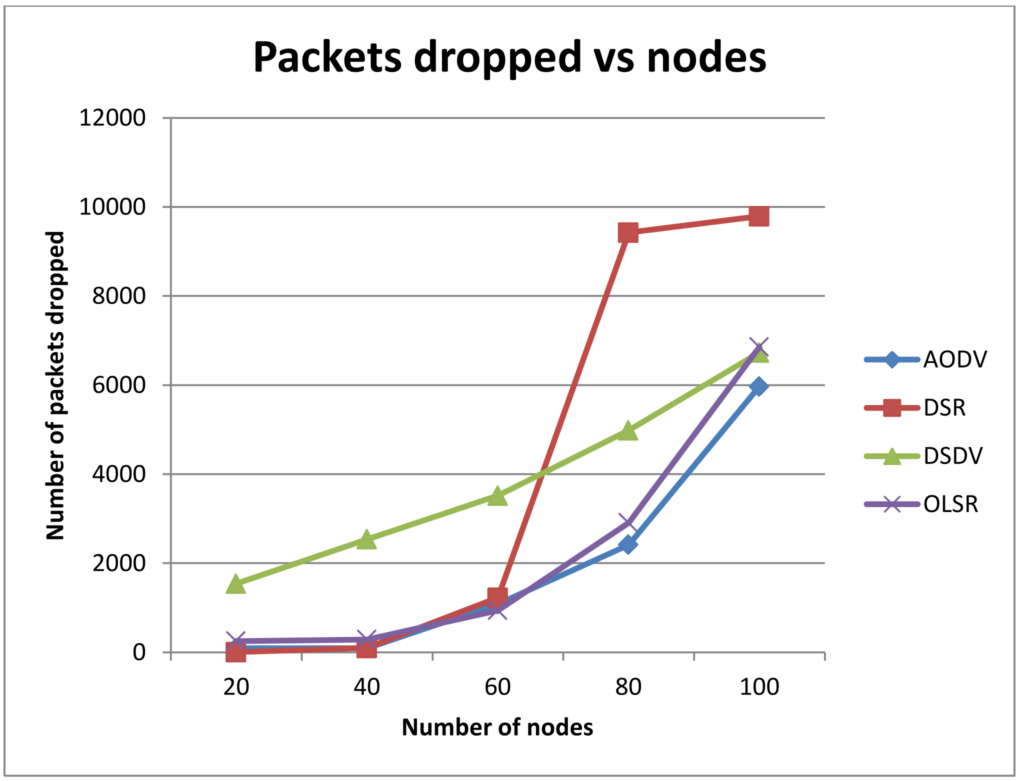

Figure 4.9 Plot of packets dropped by varying network load

4.6 Packet Analysis at maximum mobility by varying network load

In this research, a packet analysis of the various protocols is also analyzed under different traffic conditions. Table 4.4 gives a detailed analysis of the number of packets sent, received and forwarded by different when the number of nodes are varied. Maximum mobility i.e. a pause time of 0 seconds and a speed of 20 meters/second has been considered to get the below data. The number of nodes is varied from 20 to 100 which represents a small network scenario to a large network scenario. It is observed that under all the various node scenarios, the number of packets sent by the protocols are almost roughly the same - close to 3600 packets for 20 nodes, 7200 packets for 40 nodes, 10500 packets for 60 nodes, 13200 packets for 80 nodes and

Nodes		**AODV**	**DSR**	**DSDV**	**OLSR**
	Send	3602	3602	3568	3614
20	Received	3512	3599	2026	3363
	Forward	4700	4200	1599	3282
	Send	7281	7267	7306	7328
40	Received	7190	7170	4771	7040
	Forward	7767	6647	3578	5768
	Send	10500	10489	10432	10436
60	Received	9417	9265	6914	9496
	Forward	11605	16614	6317	8243
	Send	13261	13290	13226	13223
80	Received	10848	8535	8242	10312
	Forward	10171	19672	6202	7076
	Send	16213	16100	16127	16102
100	Received	10244	6310	9398	9933
	Forward	10509	29829	6504	6169

Table 4.4 Packet Analysis by varying network load

16200 packets for 100 nodes. This proves that simulation parameters that have been taken into account for the different routing protocols are accurate. Based on the routing algorithm and the efficiency of those algorithms, the received and forwarded packets are determined. This data is shown in Table 4.4. It is seen that AODV, DSR and OLSR has received almost equal number of packets till the number of nodes are 60. After 80 nodes, the DSR could not receive as many packets as AODV, thus making DSR a difficult protocol at higher network load. At 80 nodes, OLSR and AODV have received comparable number of received packets but at 100 nodes, number of packets received by AODV is more than OLSR.

DSDV shows the worse performance compared to AODV, DSR and OLSR. Compared to what has been received by the other protocols, almost 50-70 % of the packets has been received by the DSDV routing protocol when it is exposed to large scale networks. If the number of packets forwarded is analyzed, it is observed that a very high amount of packets are forwarded in DSR compared to the other protocols. This is because when the load of the network is increased, the DSR uses caching aggressively and hence a number of packets which had to be received gets forwarded.

4.7 Research observations

The simulation in this research is divided into three phases. In the first phase, the performance analysis of AODV, DSR, DSDV and OLSR is performed by varying mobility, by varying speed in the second phase and by varying the network load in the third phase. For accuracy and validation, the results of [13] are simulated and expected results are seen.

There are some significant observations that are made after this extensive simulation of the ad-hoc routing protocols. DSR performs the best when the mobility and speed of the nodes are considered. DSR performs the best out of all the protocols in high mobility and zero pause time. Even when the nodes are moving at a very high speed, performance of DSR is optimum and maintains a very low packet drop number. The reason behind this optimum performance is DSR's cache routing formula which helps DSR to maintain source routes efficiently with little time and bandwidth to maintain alternate routes. However, with the increase in the size of the network, a significant decrease in its performance is seen. When the size of the network increases, the nodes look up more to forward packets to its closest nodes. This is seen in Table 2.1 that DSR has an excessive high amount of packets forwarded, 29829 when the number of nodes is increased to 100. This pushes DSR to use caching more aggressively to creating more stale routes. These stale routes and link failures together suffice for the reason for more dropped packets. This is observed from Figure 4.9 where DSR drops more packets when the load in the network is beyond 60 nodes.

According to [9], DSDV performs the best when its plotted against simulation time. This might be only be true if the network density is minimum and the mobility is maximum. They are expected to perform better in smaller networks because of its proactive nature. When the nodes are close to stationary, DSDV has more time to update its routing table and determining the best possible route. This will lead DSDV create more stable path between the source and the destination, thereby giving a better throughput. However, it is seen how DSDV's performance fails compared to the other protocols when high mobility, speed and network density is considered, clearly stating that it is not the best protocol for a multi hop ad-hoc network.

The most important contribution of this research is that OLSR being a proactive routing protocol can be well suited in different network conditions since its performance is comparable to reactive routing protocols like AODV and DSR. It outperforms its peer, DSDV in all metrics of performance. OLSR with its MPR election strategy, provides impressive performance in small and large networks by maintaining a throughput and a packet delivery fraction close to the reactive routing protocols. With the MPR strategy, it has an advantage in slow motion environments and therefore, has a high probability of maintaining valid routes [41]. The surprisingly good observation about the OLSR protocol is that it has done well in large scale networks. Compared to AODV which should theoretically provide good performance in large networks, OLSR is a good protocol to be used in large and networks. Two main reasons - one because of its proactive nature which can be considered very stable than reactive algorithms. Second is very obvious from our observation in Table 4.4 that when the number of nodes is 100, OLSR has superior performance in forwarding packets - 6169 compared to 10509 in AODV. This is very stable considering the fact that more number of forwards might be vulnerable towards reception which happened to result in DSR's degrading performance in high density network. Moreover, the number of packets received is quite impressive - 9923 compared to 10244 in AODV. OLSR is recommended to be used in dense networks. However, previous work has stated that OLSR might have a high overhead with dense networks.

AODV has shown significant consistency in its performance being a reactive protocol. While performance of DSR has been superior in small scale networks, AODV has proved to be a better reacting protocol in loaded networks. DSR provides superior performance compared to AODV in different mobility and speed because of its caching strategy but AODV surpasses DSR's performance when the load of the network is increased. It gives the best performance out of all the routing protocols when the number of nodes is increased beyond 50. AODV also has a lower delay compared to the other routing protocols [41]. This performance can be contributed to the fact that AODV has superior knowledge of its neighbors, hence preventing loops and determining the freshest routes. Another factor that contributes to AODV's superior performance can be given to the RREQ mechanism. In DSR, the destination replies to all RREQ is receives while in AODV, it replies to only the first one it receives. Hence in high load networks, DSR finds difficulty in determining the least congested route while AODV saves all that time by using the first route that has already been preserved.

Chapter 5. Conclusions

5.1 Summary

A comprehensive performance analysis of the MANET routing protocols - AODV, DSR, DSDV and OLSR under different conditions of mobility, speed and network load is carried out in this research. DSR has the optimum performance in terms of mobility and speed and small scale networks. DSR loses its charm when the load in the network is increased. AODV has shown consistent results irrespective of the network load, speed and mobility. It fails to outperform DSR in small scale networks but maintains its superior performance even in large scale networks.

DSDV might perform good according to previous works done but it is only in small scale networks and when the mobility is minimum. In this research, it's performance has not been comparable to the other ad-hoc routing protocols.

OLSR provides an impressive performance with the matter of fact that it is a proactive routing protocol like DSDV. It has a comparable performance with AODV and has beaten DSR when the network load is high. Although it fails to cope with the level of AODV, it can be a superior protocol having demonstrated comparable performance to AODV and its proactive nature of routing packets.

All in all, DSR should be the first preference in terms of small scale networks with any mobility or speed. AODV or OLSR should be considered when the load of the network is increased. OLSR's proactive nature and comparable performance to AODV can certainly be an edge over AODV in large scale networks. However, average end-to-end delay and routing overhead between the AODV and OLSR can be a major factor to determine which stands out in large scale networks.

5.2 Future Works

There are other possible areas where this research work can be pushed. Energy consumption of the nodes in various network loads can be analyzed to track the performance analysis of the ad-hoc routing protocols. A lot of previous papers has taken into account the average end-to-end delay, routing overhead and the normalized routing load to determine the performance the ad-hoc routing protocols. Finding the optimum performance of the OLSR protocol by varying the HELLO and TC message time can also be a significant area of research. The performance analysis of the ad-hoc routing protocols can also be observed by changing the bandwidth and the transmission range of nodes and their behavior with the change of the network load.

Appendices

Appendix A

```
#
======================================================================
# NS 2 code for simulating the AODV routing protocol for MANET
#
======================================================================

#
======================================================================
# Define options
#
======================================================================
set val(chan)           Channel/WirelessChannel  ;# channel type
set val(prop)           Propagation/TwoRayGround ;# radio-propagation
model
set val(ant)            Antenna/OmniAntenna      ;# Antenna type
set val(ll)             LL                       ;# Link layer type
set val(ifq)            Queue/DropTail/PriQueue  ;# Interface queue type
set val(ifqlen)         50                       ;# max packet in ifq
set val(netif)          Phy/WirelessPhy          ;# network interface
type
set val(mac)            Mac/802_11               ;# MAC type
set val(rp)             AODV                     ;# ad-hoc routing
protocol
set val(nn)             100                       ;# number of
mobilenodes
set val(x)              1500   ;# X dimension of the topography
set val(y)              1500   ;# Y dimension of the topography
set val(seed)           1.0
set val(cp)               "./indep-utils/cmu-scen-gen/cbr-100-test"
set val(sc)               "./indep-utils/cmu-scen-gen/setdest/scen-100-
nodes"
set val(stop)           200        ;# simulation time

# Create simulator
set ns_    [new Simulator]

# Set up trace file

set tracefd  [open aodv50.tr w]      ;# for wireless traces
$ns_ trace-all $tracefd

set namtrace     [open aodv50.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)


# Create the "general operations director"
# Used internally by MAC layer: must create!
create-god $val(nn)
```

```
# Create and configure topography (used for mobile scenarios)
set topo [new Topography]
$topo load_flatgrid 1000 1000

$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0      ;# disable random motion
    $node_($i) set X_ [expr 10+round(rand()*480) ]
    $node_($i) set X_ [expr 10+round(rand()*380) ]
    $node_($i) set Z_ 0.0
}


# Define node movement model

puts "Loading connection pattern..."
source $val(cp)


# Define traffic model

puts "Loading scenario file..."
source $val(sc)


# Define node initial position in nam

for {set i 0} {$i < $val(nn)} {incr i} {

    # 20 defines the node size in nam, must adjust it according to your
      scenario
    # The function must be called after mobility model is defined

    $ns_ initial_node_pos $node_($i) 20
}


# Tell nodes when the simulation ends

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
```

```
}

$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"


puts "Starting Simulation..."
$ns_ run
$ns_ flush-trace
close $tracefd
```

Appendix B

```
#
=======================================================================
# NS 2 code for simulating the DSR routing protocol for MANET
#
=======================================================================

#
=======================================================================
# Define options
#
=======================================================================
set val(chan)          Channel/WirelessChannel  ;# channel type
set val(prop)          Propagation/TwoRayGround ;# radio-propagation
model
set val(ant)           Antenna/OmniAntenna      ;# Antenna type
set val(ll)            LL                       ;# Link layer type
set val(ifq)           CMUPriQueue  ;# Interface queue type
set val(ifqlen)        50                       ;# max packet in ifq
set val(netif)         Phy/WirelessPhy          ;# network interface
type
set val(mac)           Mac/802_11               ;# MAC type
set val(rp)            DSR                     ;# ad-hoc routing
protocol
set val(nn)            80                       ;# number of mobilenodes
set val(x)             1500    ;# X dimension of the topography
set val(y)             1500    ;# Y dimension of the topography
set val(seed)          1.0
set val(cp)              "./indep-utils/cmu-scen-gen/cbr-80-test"
set val(sc)              "./indep-utils/cmu-scen-gen/setdest/scen-80-
nodes"
set val(stop)          200        ;# simulation time

# Create simulator
set ns_     [new Simulator]

# Set up trace file
set tracefd  [open dsr50.tr w]      ;# for wireless traces
$ns_ trace-all $tracefd

set namtrace    [open dsr50.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)


# Create the "general operations director"
# Used internally by MAC layer: must create!
create-god $val(nn)


# Create and configure topography (used for mobile scenarios)
set topo [new Topography]
$topo load_flatgrid 1000 1000

$ns_ node-config -adhocRouting $val(rp) \
```

```
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0      ;# disable random motion
    $node_($i) set X_ [expr 10+round(rand()*480) ]
    $node_($i) set X_ [expr 10+round(rand()*380) ]
    $node_($i) set Z_ 0.0
}

# Define node movement model

puts "Loading connection pattern..."
source $val(cp)

# Define traffic model

puts "Loading scenario file..."
source $val(sc)

# Define node initial position in nam

for {set i 0} {$i < $val(nn)} {incr i} {

    # 20 defines the node size in nam, must adjust it according to your
      scenario
    # The function must be called after mobility model is defined

    $ns_ initial_node_pos $node_($i) 20
}


# Tell nodes when the simulation ends

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}

$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"

puts "Starting Simulation..."
$ns_ run

$ns_ flush-trace
close $tracefd
```

Appendix C

```
#
=======================================================================
# NS 2 code for simulating the DSDV routing protocol for MANET
#
=======================================================================

#
=======================================================================
# Define options
#
=======================================================================
set val(chan)          Channel/WirelessChannel  ;# channel type
set val(prop)          Propagation/TwoRayGround ;# radio-propagation
model
set val(ant)           Antenna/OmniAntenna      ;# Antenna type
set val(ll)            LL                       ;# Link layer type
set val(ifq)           Queue/DropTail/PriQueue  ;# Interface queue type
set val(ifqlen)        50                       ;# max packet in ifq
set val(netif)         Phy/WirelessPhy          ;# network interface
type
set val(mac)           Mac/802_11               ;# MAC type
set val(rp)            DSDV                     ;# ad-hoc routing
protocol
set val(nn)            20                       ;# number of mobilenodes
set val(x)             1500   ;# X dimension of the topography
set val(y)             1500   ;# Y dimension of the topography
set val(seed)          1.0
set val(cp)              "./indep-utils/cmu-scen-gen/cbr-20-test"
set val(sc)              "./indep-utils/cmu-scen-gen/setdest/scen-20-
nodes"
set val(stop)          200        ;# simulation time

# Create simulator
set ns_     [new Simulator]

# Set up trace file
set tracefd  [open dsdv50.tr w]      ;# for wireless traces
$ns_ trace-all $tracefd

set namtrace    [open dsdv50.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)


# Create the "general operations director"
# Used internally by MAC layer: must create!
create-god $val(nn)


# Create and configure topography (used for mobile scenarios)
set topo [new Topography]
$topo load_flatgrid 1000 1000

$ns_ node-config -adhocRouting $val(rp) \
```

```
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0       ;# disable random motion
    $node_($i) set X_ [expr 10+round(rand()*480) ]
    $node_($i) set X_ [expr 10+round(rand()*380) ]
    $node_($i) set Z_ 0.0
}

# Define node movement model

puts "Loading connection pattern..."
source $val(cp)


# Define traffic model

puts "Loading scenario file..."
source $val(sc)

# Define node initial position in nam

for {set i 0} {$i < $val(nn)} {incr i} {

    # 20 defines the node size in nam, must adjust it according to your
      scenario
    # The function must be called after mobility model is defined

    $ns_ initial_node_pos $node_($i) 20
}

# Tell nodes when the simulation ends

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}

$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
puts "Starting Simulation..."
$ns_ run

$ns_ flush-trace
close $tracefd
```

Appendix D

```
#
========================================================================
# NS 2 code for simulating the OLSR routing protocol for MANET
#
========================================================================

#
========================================================================
# Define options
#
========================================================================
set val(chan)          Channel/WirelessChannel  ;# channel type
set val(prop)          Propagation/TwoRayGround ;# radio-propagation
model
set val(ant)           Antenna/OmniAntenna      ;# Antenna type
set val(ll)            LL                       ;# Link layer type
set val(ifq)           CMUPriQueue  ;# Interface queue type
set val(ifqlen)        50                       ;# max packet in ifq
set val(netif)         Phy/WirelessPhy          ;# network interface
type
set val(mac)           Mac/802_11               ;# MAC type
set val(rp)            OLSR                     ;# ad-hoc routing
protocol
set val(nn)            80                       ;# number of mobile
nodes
set val(x)             1500   ;# X dimension of the topography
set val(y)             1500   ;# Y dimension of the topography
set val(seed)          1.0
set val(cp)              "./indep-utils/cmu-scen-gen/cbr-80-test"
set val(sc)              "./indep-utils/cmu-scen-gen/setdest/scen-80-
nodes"
set val(stop)          200        ;# simulation time

# Create simulator
set ns_     [new Simulator]
Agent/OLSR set use_mac_    true

# Set up trace file
set tracefd  [open olsr50.tr w]      ;# for wireless traces
$ns_ trace-all $tracefd

set namtrace    [open olsr50.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)


# Create the "general operations director"
# Used internally by MAC layer: must create!
create-god $val(nn)


# Create and configure topography (used for mobile scenarios)
set topo [new Topography]
$topo load_flatgrid 1000 1000
```

```
$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -channel [new $val(chan)] \
    -topoInstance $topo \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0      ;# disable random motion
    $node_($i) set X_ [expr 10+round(rand()*480) ]
    $node_($i) set X_ [expr 10+round(rand()*380) ]
    $node_($i) set Z_ 0.0
}


# Define node movement model

puts "Loading connection pattern..."
source $val(cp)


# Define traffic model

puts "Loading scenario file..."
source $val(sc)


# Define node initial position in nam

for {set i 0} {$i < $val(nn)} {incr i} {

    # 20 defines the node size in nam, must adjust it according to your
      scenario
    # The function must be called after mobility model is defined

    $ns_ initial_node_pos $node_($i) 20
}


# Tell nodes when the simulation ends

for {set i 0} {$i < $val(nn) } {incr i} {
    $ns_ at $val(stop).0 "$node_($i) reset";
}

$ns_ at  $val(stop).0002 "puts \"NS EXITING...\" ; $ns_ halt"
```

```
puts "Starting Simulation..."
$ns_ run

$ns_ flush-trace
close $tracefd
```

Appendix E

```
#
=======================================================================
# A sample connection pattern file with 20 nodes
#
=======================================================================



#
# nodes: 20, max conn: 6, send rate: 0.25, seed: 1.0
#
#
# 1 connecting to 2 at time 2.5568388786897245
#
set udp_(0) [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp_(0)
set null_(0) [new Agent/Null]
$ns_ attach-agent $node_(2) $null_(0)
set cbr_(0) [new Application/Traffic/CBR]
$cbr_(0) set packetSize_ 512
$cbr_(0) set interval_ 0.25
$cbr_(0) set random_ 1
$cbr_(0) set maxpkts_ 10000
$cbr_(0) attach-agent $udp_(0)
$ns_ connect $udp_(0) $null_(0)
$ns_ at 2.5568388786897245 "$cbr_(0) start"
#
# 4 connecting to 5 at time 56.333118917575632
#
set udp_(1) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(1)
set null_(1) [new Agent/Null]
$ns_ attach-agent $node_(5) $null_(1)
set cbr_(1) [new Application/Traffic/CBR]
$cbr_(1) set packetSize_ 512
$cbr_(1) set interval_ 0.25
$cbr_(1) set random_ 1
$cbr_(1) set maxpkts_ 10000
$cbr_(1) attach-agent $udp_(1)
$ns_ connect $udp_(1) $null_(1)
$ns_ at 56.333118917575632 "$cbr_(1) start"
#
# 4 connecting to 6 at time 146.96568928983328
#
set udp_(2) [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp_(2)
set null_(2) [new Agent/Null]
$ns_ attach-agent $node_(6) $null_(2)
set cbr_(2) [new Application/Traffic/CBR]
$cbr_(2) set packetSize_ 512
$cbr_(2) set interval_ 0.25
$cbr_(2) set random_ 1
$cbr_(2) set maxpkts_ 10000
```

```
$cbr_(2) attach-agent $udp_(2)
$ns_ connect $udp_(2) $null_(2)
$ns_ at 146.96568928983328 "$cbr_(2) start"
#
# 6 connecting to 7 at time 55.634230382570173
#
set udp_(3) [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp_(3)
set null_(3) [new Agent/Null]
$ns_ attach-agent $node_(7) $null_(3)
set cbr_(3) [new Application/Traffic/CBR]
$cbr_(3) set packetSize_ 512
$cbr_(3) set interval_ 0.25
$cbr_(3) set random_ 1
$cbr_(3) set maxpkts_ 10000
$cbr_(3) attach-agent $udp_(3)
$ns_ connect $udp_(3) $null_(3)
$ns_ at 55.634230382570173 "$cbr_(3) start"
#
# 7 connecting to 8 at time 29.546173154165118
#
set udp_(4) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(4)
set null_(4) [new Agent/Null]
$ns_ attach-agent $node_(8) $null_(4)
set cbr_(4) [new Application/Traffic/CBR]
$cbr_(4) set packetSize_ 512
$cbr_(4) set interval_ 0.25
$cbr_(4) set random_ 1
$cbr_(4) set maxpkts_ 10000
$cbr_(4) attach-agent $udp_(4)
$ns_ connect $udp_(4) $null_(4)
$ns_ at 29.546173154165118 "$cbr_(4) start"
#
# 7 connecting to 9 at time 7.7030203154790309
#
set udp_(5) [new Agent/UDP]
$ns_ attach-agent $node_(7) $udp_(5)
set null_(5) [new Agent/Null]
$ns_ attach-agent $node_(9) $null_(5)
set cbr_(5) [new Application/Traffic/CBR]
$cbr_(5) set packetSize_ 512
$cbr_(5) set interval_ 0.25
$cbr_(5) set random_ 1
$cbr_(5) set maxpkts_ 10000
$cbr_(5) attach-agent $udp_(5)
$ns_ connect $udp_(5) $null_(5)
$ns_ at 7.7030203154790309 "$cbr_(5) start"
#
#Total sources/connections: 4/6
#
```

Appendix F

```
#
======================================================================
# AWK script to calculate the Throughput of the network
#
======================================================================


BEGIN {

            recvdSize = 0
            startTime = 400
            stopTime = 0

    }


    {

               event = $1
               time = $2
               node_id = $3
               pkt_size = $8
               level = $4



           # Store start time

           if (level == "AGT" && event == "s" && pkt_size >= 512)
        {

            if (time < startTime) {
                    startTime = time
                    }

            }



          # Update total received packets' size and store packets
arrival                 time

           if (level == "AGT" && event == "r" && pkt_size >= 512)
        {
                if (time > stopTime) {
                        stopTime = time
                    }

                # Rip off the header

                hdr_size = pkt_size % 512
                pkt_size -= hdr_size
```

```
            # Store received packet's size

        recvdSize += pkt_size
        throughput = (recvdSize/(stopTime-startTime))*(8/1000)

        #printf("Average Throughput[kbps] = %.2f\t\t

  StartTime=%.2f\tStopTime=%.2f\n",throughput,startTime,stopTime)

        printf("%f\n", throughput) >"throughput_plot_trace";
        #printf("%f %f\n", time, throughput)


        }

 }



END {

 #printf("Done")
}
```

Appendix G

```
#
========================================================================
# AWK script to calculate the number of sent packets, received packets,
forwarded packets, dropped packets and Packet delivery Ratio of the
network
#

========================================================================

BEGIN {

            sendLine = 0;

            recvLine = 0;

            fowardLine = 0;

    }


    $0 ~/^s.* AGT/ {

            sendLine ++ ;

    }a


   $0 ~/^r.* AGT/ {

            recvLine ++ ;

   }


   $0 ~/^f.* RTR/ {

            fowardLine ++ ;

   }


 END {

           printf "cbr send:%d received:%d, PDF:%.4f, forwarded:%d,
dropped:%d \n", sendLine, recvLine, (recvLine/sendLine),fowardLine,
(sendLine-recvLine);

   }
```

References

1. J.Hoebeke, I. Moerman, B. Dhoedt, P. Demeester, "An overview of mobile ad hoc networks: Applications and challenges", Journal-Communications Network 3.3 (2004): 60-66.

2. http://adrianlatorre.com/projects/pfc/img/vanet_full.jpg

3. http://www.cse.wustl.edu/~jain/cis788-99/ftp/adhoc_routing/#TDRP

4. http://www.cse.wustl.edu/~jain/cis788-99/ftp/adhoc_routing/#DSDV

5. F.Bertocci, P.Bergamo, G.Mazzini, M.Zorzi, "Performance Comparison of Routing Protocols for Ad Hoc networks", IEEE Global Telecommunications Conference (GLOBECOM) 2003

6. G.Adam, V.Kapoulas, C.Bouras, G.Kioumourtzis, A.Gkamas, N.Tavoularis, "Performance Evaluation of Routing Protocols for multimedia transmission over Mobile Ad Hoc Networks", Wireless and Mobile Networking Conference (WMNC) 2011.

7. S. Das, C. Perkins, E. Royer, "Performance Comparison of two on-demand routing protocols for ad-hoc networks", IEEE International Conference on Computer Communications (INFOCOM) 2000.

8. S.A. Ade and P.A Tijare, "Performance comparison of AODV, DSDV, OLSR and DSR routing protocols in mobile ad-hoc networks", International Journal of Information Technology and Knowledge management, Jul-Dec 2010

9. Md. S. Islam, Md. N. Hider, Md. T. Haque, L.Miah, "An extensive comparison

among DSDV, DSR and AODV protocols in MANET, International Journal of Computer Applications, Feb-2011

10. H.Ehsan, Z. Uzmi, "Performance comparison of adhoc wireless network routing protocols", Proceedings of IEEE International Multi-topic Conference (INMIC) 2004

11. V. Talooki and K. Ziarati, "Performance comparison of routing protocols for mobile adhoc networks", Asia-Pacific Conference on Communications, APCC, August-September 2006

12. Md. Morshed et all, "Performance comparison of TCP variants over AODV, DSDV, DSR, OLSR in NS-2", IEEE International conference on Informatics, Electronics and Vision, 2012

13. Payal, S. Jakhar, "CBR Traffic Based Performance Investigations of DSDV, DSR and AODV Routing Protocols for MANET Using NS2", International Journal of Soft Computing and Engineering, ISSN: 2331-2307, Volume-3, Issue-4, September 2013

14. Q Feng, Z. Cai, J. Yang, X. Hu "A Performance Comparison of the Ad Hoc Network Protocols", Second International Workshop on Computer Science and Engineering, 2009

15. S. Hosseini, H. Farrokhi, " The Impacts of Network size on the performance of routing protocols in mobile ad-hoc networks", Second Pacific-Asia conference on Circuits, Communications and System (PACCS), 2010

16. P. Liu, D. Chen, C. Hu, W. Sun, J. Lee, C. Chou, W. Shih, "Analyzing the TCP

performance on mobile ad-hoc networks", 13th International Conference on Advanced Communications Technology (ICACT), 2011

17. C.Mbarushimana, A.Shahrabi, "Comparative Study of reactive and proactive routing protocols performance in mobile ad-hoc networks", 21st International Conference on Advanced Information Networking and Applications workshop (AINAW), 2007

18. B. Gouda, A. Das, K. Lakshmi, "A Comprehensive performance analysis of energy efficient routing protocols in different traffic based mobile ad-hoc networks", International multi-conference on automation, computing, communications, control and compressed sensing, 2013

19. P. Gupta, S. Gupta, "Performance Evaluation of mobility models on MANET routing protocols", Advanced Computing and Communications Technologies (ACCT), 2013

20. B.S Gouda, A.K Dass, K.L. Narayana, " A comprehensive performance analysis of energy efficient routing protocols in different traffic based mobile ad-hoc networks", International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013

21. M.Morshed, M. Rahman, M. Islam, " Performance comparison of TCP variants over AODV, DSDV, DSR, OLSR in NS-2", International Conference on Informatics, Electronics & Vision (ICIEV), 2012

22. M. Rajeswari, P. Maheswari, S. Bhuvaneswari, " Performance analysis of AODV, DSR, TORA and OLSR to achieve group communication in MANET", Fourth

International Conference on Advanced Computing (ICoAC), 2012

23. K. Natarajan, G. Mahadevan, " A Succinct Comparative Analysis and Performance of MANET routing protocols", International Conference on Computer Communication and Informatics (ICCCI -2013), Jan. 04 – 06, 2013

24. P. Chauhan, S. Vijay, P. Arya, "Comparative Analysis of Routing Protocols in AD-HOC Network: AODV, DSDV, DSR", International Journal of Current Engineering and Technology, 2013

25. S. Bhujade, S. Sawant, " Evaluation of AODV,DSR and DSDV using NS2", International Journal of Engineering Trends and Technology (IJETT), 2013

26. P. Gaharwar, S. Gupta, "Performance comparison of routing protocols", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 4, April 2013

27. Thirveni H.B, M. Kumar, R. Sharma, "Performance Evaluation of Routing Protocols in Mobile Ad-Hoc Networks with Varying Node Density and Node Mobility", International Conference on Communication Systems and Network Technologies, 2013

28. N. Pal, R. Dhir, " Analyze the Impact of Mobility on Performance of Routing Protocols in MANET Using OPNET Modeller", International Journal of Advanced Research in Computer Science and Software Engineering 2013

29. P. Meher, P. Kulkarni, "Analysis and Comparison of Performance of TCP-Vegas in MANET", International Conference on Communication Systems and Network Technologies, 2011

30. R. Jain, N. Khairner, L. Shrivastava, "Comparative Study of Three Mobile Ad-hoc Network Routing Protocols under Different Traffic Source", International Conference on Communication Systems and Network Technologies, 2011

31. L. Shrivastava, S. Bhaduria, G. Tomar, "Performance Evaluation of Routing Protocols in MANET with different traffic loads", International Conference on Communication Systems and Network Technologies 2011

32. P. Rani, N. Sharma, P. Singh, " Performance Comparison of VANET Routing Protocols", International Conference on Wireless Communications, Networking and Mobile Computing, 2011

33. M. Rajeswari, U. Maheswari, S. Bhuvaneswari, " Performance analysis of AODV, DSR, TORA and OLSR to achieve group communication in MANET", Fourth International Conference on Advanced Computing, ICoAC 2012

34. T. Arefin, T. Khan, I. Toyoda, " Performance Analysis of Mobile Ad-hoc Network Routing Protocols", IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision, 2012

35. Fu Yongsheng, Wang Xinyu, Li Shanpin, "Performance comparison and analysis of routing strategies in Mobile ad hoc networks", International Conference on Computer Science and Software Engineering, 2008

36. Sabina Baraković, Suad Kasapović, and Jasmina Baraković, "Comparison of MANET Routing Protocols in Different Traffic and Mobility Models", Telfor Journal, Vol. 2, No. 1, 2010.

37. Dr. C. Putta, Dr. B. Prasad, D. Ravilla, M. Nath, M. Chandra, "Performance of Ad

hoc Network Routing Protocols in IEEE 802.11", Int’l Conf. on Computer & Communication Technology, 2010

38. A. Kumar, L. Reddy, P. Hiremath, " Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6, June 2008

39. R. Noorani, " Comparative Analysis of Reactive MANET Routing Protocols Under the Traffic of TCP VEGAS with Mobility Considerations", International Conference on Emerging Technologies, 2009

40. S. Kumar, R. Rathy, D. Pandey, "Traffic Pattern Based Performance Comparison of Two Reactive Routing Protocols for Ad Hoc Networks Using NS2", International Conference on Computer Science and Information Technology, 2009. ICCSIT

41. P. Rani, N. Sharma, P. Singh, "Performance comparison of VANET Routing protocols", International Conference on Wireless Communications, Networking and Mobile Computing (WiCom), 2011

42. http://www.isi.edu/nsnam/ns/tutorial/